Tree Structure in Phylogenetic Networks

J. R. Simpson

A thesis submitted in partial fulfilment of the requirements for the
Degree of
Doctor of Philosophy in Mathematics

Supervised by Professor Charles Semple and Professor Mike Steel
School of Mathematics and Statistics
University of Canterbury
New Zealand
October 7, 2019

Abstract

Phylogenetic trees are widely used to express and explore evolutionary relationships. In recent times, the observation of evolutionary processes that cannot be expressed by individual phylogenetic trees has prompted interest in the study of phylogenetic networks. Phylogenetic networks generalise phylogenetic trees by allowing non-treelike events to be represented. A particular consequence of this is that a phylogenetic network may be understood to simultaneously express the relationships of a number of different phylogenetic trees. These phylogenetic trees are then said to be embedded in the network.

In this thesis, the connections between various classes of phylogenetic networks and their corresponding sets of embedded phylogenetic trees are explored. Among others, the following questions are expanded on and answered.

1. For a given set of trees does there exist a network that embeds each tree? In the case of level-1 networks a polynomial time algorithm is given that outputs, up-to a particular topological ambiguity, the unique level-1 network with minimum reticulations that displays a given set of trees or identifies that no such network exists.

2. From a given set of trees embedded in a network can the network be reconstructed? It is proven that a normal network can be reconstructed from a subset of the trees it displays that grows linearly with respect to the number of leaves in the network.

3. For a given network how many embedded trees are required to use every vertex and every edge of the network? It is proven that the class of stack-free network is precisely the class of networks for which only two embedded trees are required to use every vertex and every edge of the network.

4. For a given network and tree does the network embed the tree? In the case of sibling-free networks a polynomial time algorithm is given that outputs, for a given network and tree, whether or not the network embeds the tree.

Acknowledgements

To begin with, I would like to acknowledge the tremendous support I have received during my time as a Ph.D. student. First and foremost, I would like to thank my supervisors Charles Semple and Mike Steel. They not only supported and directed my research, lending me their considerable experience and insight but most importantly, on faith, they gave me this opportunity to study towards a Ph.D. As someone who has always struggled with learning disabilities I will always be grateful for this belief. In conjunction with my study I again have to thank Charles and Mike for funding my attendance of a two week course on algebraic and combinatorial phylogenetics hosted by the Barcelona Graduate School of Mathematics as well as two of the annually held New Zealand Phylogenomics Meetings. From these I was able to meet and learn from world leading experts in mathematical phylogenetics. I also have to thank the University of Canterbury School of Mathematics and Statistics for funding my attendance of five of the annually held New Zealand Mathematics and Statistics Postgraduate Conferences and two of the annually held New Zealand Mathematics Colloquiums. These conferences allowed me to meet other Ph.D. students from all over New Zealand and practice formally presenting my research. Next I would like to thank the University of Canterbury for awarding me the University of Canterbury Doctoral Scholarship for Students with Disabilities to financially support me in my final year. Finally I would like to thank my friends, family and partner for all of their support, encouragement and unfailing patience.

Contents

1 Introduction **5**

2 Preliminaries **14**

3 Level-1 Network Construction **24**

3.1 Introduction 24

3.2 Preliminaries 28

3.3 Level-1 Uniqueness 31

3.4 Level-1 Construction 36

3.5 Summary 46

4 Tree-Child Network Construction **47**

4.1 Introduction 47

4.2 Preliminaries 49

4.3 Tree-Child Display vs Level-1 Display 52

4.4 What Cannot be Displayed by a Tree-child Network . . . 59

4.5 Summary 65

5 Normal Network Reconstruction **66**

5.1 Introduction 66

5.2 Preliminaries 67

5.3 Normal Network Reconstruction 70

5.4 Summary 78

6 Beyond Tree-child Networks **80**

6.1 Introduction 80

6.2 Preliminaries 83

6.3 Stack-Free Networks 86

6.4 Questions for Stack-free Networks 92

6.5 Counting the Trees that Cover a Network 102

6.6 Sibling-Free Networks 109

6.7 Summary 116

7 Conclusion **118**

Chapter 1

Introduction

Phylogenetics is 'the theory of reconstructing and analysing trees or more complex networks from data observed in the present' [Ste16]. This is typically but not exclusively understood in the context of evolutionary biology. For example, phylogenetics is also used in computer science [All70], linguistics [AG05] and stemmatology [HBMR04]. Simple, clear and exact, phylogenetic trees have long been used to illustrate the relationships between present-day species and their hypothetical common ancestors.

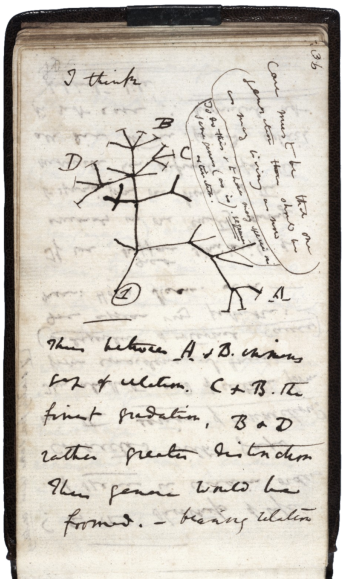

Figure 1.1: Darwin's Tree.
An early phylogenetic tree sketched by Charles Darwin in 1837 [Dar37].

In much the same way as a family tree, a phylogenetic tree represents the relationships of parent and child, ancestor and descendant and siblings between different present-day and hypothetical species detailing their evolutionary history. The use of phylogenetic trees in evolutionary biology is readily traced back to the time of Charles Darwin, as can be seen in Fig. 1.1. Importantly, phylogenetic trees have provided an intuitive and eloquent way of expressing these relationships, greatly facilitating their study and exploration.

Simply put, a phylogenetic tree is a picture consisting of dots and lines, for example see Fig. 1.2. The dots, known as vertices represent discrete items such as present-day and the hypothetical ancestors of present-day species. The lines, known as arcs or edges each join two vertices and indicate the direction from parent to child. The final descendants in a tree are vertices with no outgoing edges. These vertices are called leaves and are labelled or assumed to be identifiable. They represent known items where, in contrast, the other vertices represent the hypothetical ancestors of the leaves. Every vertex in a phylogenetic tree may have at most one in-coming edge. As a consequence of this restriction, phylogenetic trees have particularly useful properties.

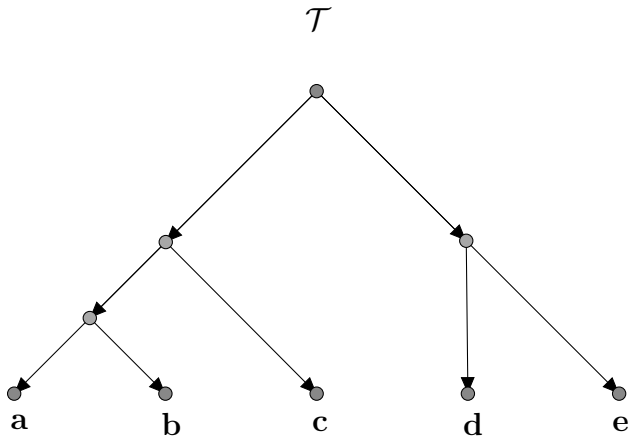

Figure 1.2: Phylogenetic Network.
Here is an example of a phylogenetic network on leaves a, b, c, d and e.

Presuming that phylogenetic trees can represent the different evolutionary relationships that occur in nature, their underlying mathematical properties may be exploited. One specific example is that a phylogenetic tree is uniquely determined by a dense set of rooted triples [ASSU81]. Because of this property, which is formally defined in Chapter 3, for an arbitrarily large set of leaves it is sufficient to find the relative relationship between subsets of only three leaves at a time to produce, if it is possible, a phylogenetic tree that represents all of the relationships between every single leaf in the entire set. More importantly, the phylogenetic tree produced this way is the only phylogenetic tree that can represent these relationships between these leaves. This means that if for every combination of three living species, two could be found that are more closely related to each other than the third then, if it exists, the only possible phylogenetic tree that represents the relationship of all life can be found. This would be the true tree, the so-called Tree of Life.

The Tree of Life is the idea of a grand unifying phylogenetic tree that details the evolutionary relationship of all life, past and present. An illustration of what this might look like is given in Fig. 1.3. If the Tree of Life could be constructed important and broad questions about the evolutionary history of all life could be directly explored. Moreover, progress in working with DNA and improving computing power hint that the construction of the Tree of Life could soon be possible and consequently its secrets laid bare. However, it is here worth pausing to recall that this story has been told before. It was the Tree of Life that Odin hung himself from in sacrifice to understand the magic of the universe, the Buddha sat under to receive enlightenment and was sealed in the Garden of Eden forbidden from mankind. The Tree of Life is a archetypical and evocative idea but it is wrong to suppose a priori that nature conforms to it.

As our understanding of genetics has developed it has become increasingly more evident that non-treelike evolution plays a significant roll in nature. Explained by W. Ford Doolittle, “Molecular phylogeneticists will have failed to find the ‘true tree’, not because their methods are inadequate or because they have chosen the wrong genes, but because the history of life cannot be properly represented as a tree” [Doo99].

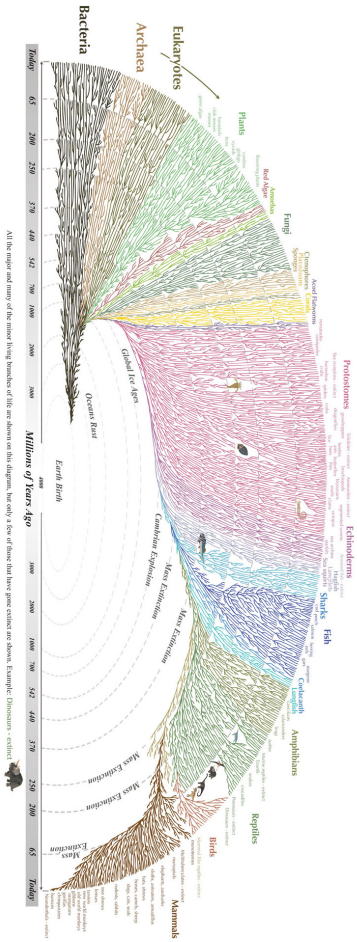

Figure 1.3: Tree of Life
Image of The Tree of Life [Eis08].

These non-treelike events include horizontal gene transfer, recombination and hybridization. Horizontal gene transfer is the transfer of genes between species, recombination is the exchange of genetic material on chromosomes and hybridization is the interbreeding of what is typically viewed as different species. In order for such events to be represented, vertices must be allowed to have more than one edge directed towards them or a way must be found to choose which of the potential incoming edges is the most important. The former case generalises phylogenetic trees into phylogenetic networks. Phylogenetic networks have both tree-vertices, vertices with at most one incoming edge, and reticulation vertices, vertices with at least two incoming edges. For example see the phylogenetic network $\mathcal{N}$ with reticulation-vertex r in Fig. 1.4.

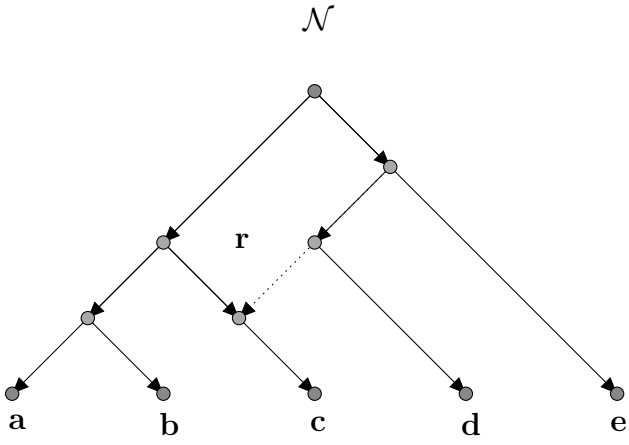

Figure 1.4: Phylogenetic Network
A phylogenetic network on the leaves a, b, c, d and e and a reticulation-vertex r.

Note, if the two incoming edges on the reticulation-vertex r are considered separately, two different phylogenetic trees are represented, one where the leaf c is more closely related to a and b and the other where it is more closely related to d and e. These two trees are said to be embedded trees in the phylogenetic network $\mathcal{N}$.

Typically centred on prokaryotes, simple single-celled organisms where most non-treelike evolution is observed, non-treelike events prompt debate as to whether or not the idea of a tree has any utility at all. In contrast to the view of Doolittle, Eugene Koonin argues that the " 'tree of life' remains a cornerstone of evolutionary biology although it has to be re-conceptualized" [Koo15]. This issue is foundational. As put by the linguistic philosopher Ludwig Wittgenstein "whereof one cannot speak, thereof one must be silent" [Wit13]. Without definitive means to represent and express evolutionary relationships, any subsequent analysis or research is at best handicapped.

It is important to note that it is because of the useful properties of trees that they are employed in the first place. Consider particularly how phylogenetic trees both represent relationships in a clear and intuitive way and provide a rigid structure that gives many problems the necessary tractability to be solved efficiently. In fact the natural utility of trees goes so far as to stunt non-treelike phylogenetic research.

Much non-treelike evolutionary behaviour has until recently been "overlooked" because of the "deep-rooted expectation of treelike evolution" that has ingrained itself over the decades [BvIJ+13]. An example of this was pointed out by Dagan and Martin who observed attempts to fit data to a tree resulting in phylogenetic trees that represent the evolution of only one per cent of the genes involved [DM06].

The challenge of replacing or adapting the Tree of Life is then not as simple as adopting an object that can express all observable evolutionary events, treelike and non-treelike. Any phylogenetic network that allows branches to recombine arbitrarily could be used to serve this purpose. However, with no guiding structure many pertinent questions would be then left either entirely intractable or with a number of solutions and no obvious way of choosing between them. Furthermore, it is generally believed that the evolution of individual genes does occur in a treelike manner (see [VISS10], [CLS14], [BvIJ+13]). This would mean trees are relevant at a fundamental level and the way in which trees may occur 'in' any such object is critical.

Three distinct methods have been pursued to expand beyond the use of simple phylogenetic trees. First is the 'statistical tree of life', where probabilistic tools are used to find amongst conflicting trees on the same leaf set which one is most likely, reducing the others to noise [Koo15]. Second is deep coalescences, where gene trees are allowed to exist inside of the superstructure of a species tree, see Fig. 1.5. Third is phylogenetic networks, where the structure of a tree is generalised to allow vertices to have more than one edge directed towards them. If for a given set of species different phylogenetic trees were produced depending on what genes were sampled then all three methods could conceptually resolve the conflicting data and express a possible evolutionary history for the set of species. However, the former two methods still assume evolution to be fundamentally treelike whilst dismissing the clarity and precision of a simple tree.

Ultimately it might be expected that a combination of these methods is required. This would be to transition phylogenetic thinking from the expectation of rigid and precise relationships to more expressive modelling where additional complexity is gradually added. An example of

such a combination of methods is multi-species network coalescences [DDBR09]. Still, only the use of phylogenetic networks allow evolutionary data to be explored without the implicit bias towards treelike structures. Because of this phylogenetic networks have been extensively studied over the past twenty years. Of particular interest are the consequences of specific structural restrictions on a network with regards to what and how different trees can be combined and expressed.

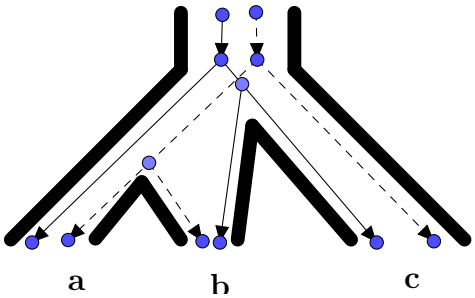

Figure 1.5: Deep Coalescences
Here is an example of how a deep coalescences model allows contradicting gene trees (the solid lined tree and the dashed lined tree) to coexist inside the superstructure of a single species tree (the tree existing between the solid bold lines).

The central aim of this thesis is to examine the underlying structural properties that affect the expression of conflicting sets of trees for different classes of phylogenetic networks. This works from the premise that a phylogenetic network exists as an amalgamation of precisely determinable phylogenetic trees. Phylogenetic networks have frequently been understood in this way since at least 1990 in [Hei90] . This is approached from two different perspectives. The first is to ask the reverse of the well-studied tree-containment problem (see [GVIK$^+$16], [KNTX08], [VISS10]). The tree-containment problem asks if a given phylogenetic tree is embedded in a given phylogenetic network. Here, it is asked if for a particular class of phylogenetic networks and a given set of phylogenetic trees, is there a phylogenetic network that embeds each tree in the set? This will be called the *network-capture problem*. The second is to ask for a particular class of phylogenetic networks, what is the minimum number of embedded phylogenetic trees required to use every vertex and every edge of the network? Here this problem will be called the *tree-cover problem*.

Overview

This thesis is organised as follows. In the next chapter the initial definitions and already known results are formally detailed and explained. In Chapter 3 the network-capture problem is explored for the rigidly structured class of level-1 phylogenetic networks. Chapter 4 and Chapter 5 expand the network-capture problem to the more general class of tree-child networks. Chapter 6 considers the tree-cover problem and tree-containment problem for several classes of phylogenetic networks that generalise tree-child networks. Then finally the results of this thesis are summarised in the conclusion along with potential questions for further research. Outside of the preliminary chapter and each preliminary section, unless otherwise stated, all propositions and theorems in this work are believed to be new results. Inside the preliminary chapter and sections all results are believed to be known but unless otherwise stated could not be found published in the literature.

In Chapter 3 the rigidly structured nature of level-1 networks is exploited to introduce and flesh out the network capture problem. Here, two important properties are identified. First, a level-1 network that embeds two trees so that every incoming edge on a reticulation-vertex is used by an embedding of one of the two trees, uses the minimum possible number of reticulation vertices to embed the two trees. Second, a level-1 network that embeds a set of trees with the minimum necessary number of reticulation vertices is the only level-1 network, up-to a particular topological ambiguity, that embeds that set of trees with that number of reticulation vertices. These two properties are then used to produce a polynomial time algorithm that for a given set of trees, either constructs a level-1 network with the minimum necessary reticulation vertices that embeds the set or identifies that no such network exists.

In Chapter 4 the network capture problem is considered for the more general class of tree-child networks. To begin, the first of the two earlier mentioned properties for level-1 networks is shown not to hold for tree-child networks. Instead, there can exist a tree-child network that embeds two trees so that every incoming edge on a reticulation-vertex is used by an embedding of one of the two trees and another tree-child network that does the same with less reticulation vertices.

This makes it difficult to apply the strategy used for level-1 networks to tree-child networks. Next, it is demonstrated that there exists a set of only three trees that cannot be displayed by any tree-child network. The attributes of this set of trees that prevent it from being displayed are then generalised so as to be readily identifiable in any given set of trees.

In Chapter 5 the network capture problem is re-examined for tree-child networks with the benefit of two assumptions. This time, in contrast to the strategy used for level-1 networks, the elements in the set of given trees are considered simultaneously. It is assumed that first, for the given set of trees there exists a tree-child network that embeds each tree in the set and second, this network contains no short-cut edges. With the tractability provided by these two assumptions it is proven that there exists a polynomial time algorithm that from a given special subset of trees constructs the unique such network that embeds this set of trees.

In Chapter 6 the tree-cover problem and tree-containment problem are considered for classes of phylogenetic networks that generalise tree-child networks. The work in this chapter is based on the paper [SS18] that was jointly written by Charles Semple and myself. For the tree-cover problem, it is identified that the class of stack-free networks, networks where no reticulation is the parent of another, is precisely the class of phylogenetic networks for which two special embedded trees in a network use every vertex and every edge of the network. Based on this the following results are shown. A polynomial time algorithm that identifies if a given phylogenetic network has two embedded trees that use every vertex and every edge of the network. A new characterisation of the class of reticulation-visible networks. There exists a universal network, a network that embeds every tree on the same leaf set, that has two embedded trees that use every vertex and every edge of the network. A polynomial time algorithm that decides if two given phylogenetic trees are embedded in a particular subclass of reticulation-visible networks and together use every vertex and every edge of the network. Then finishing this chapter, the tree-cover problem is answered for the class of sink-visible networks and the tree-containment problem is shown to be solvable in polynomial time for sibling-free networks.

Chapter 2

Preliminaries

In this chapter the formal background necessary to begin this work is given. Here, the terms relevant throughout the thesis are defined and the results, known in the literature, that this work builds on are introduced and independently proven. A basic familiarity with graph theory is assumed. For a wider and more detailed perspective of the terms and ideas discussed in this work see [Ste16].

Directed graphs. The phylogenetic networks considered in this work are a special type of directed graphs. A *directed graph* G consists of a vertex set $V(G)$ and an edge set $E(G)$. Each element uv of $E(G)$ joins two distinct elements u and v of $V(G)$ and indicates a direction going from the *parent vertex* u to the *child vertex* v. The edge uv will be referred to as an *outgoing edge* of u and an *incoming edge* of v. The *out-degree* of a vertex is the number of outgoing edges from that vertex and similarly the *in-degree* of a vertex is the number of incoming edges to that vertex. The *degree* of a vertex is the sum of its in-degree and out-degree.

The internal aspects of a graph are understood in terms of paths. A *path* in G is a sequence of vertices $u_1, u_2, ..., u_k$ where all but the first and last elements must be distinct, connected by a sequence of distinct edges that join each subsequent vertex and the one preceding it. The *length of a path* is the number of edges that connect the vertices of the path. The two main types of paths used in this work are directed paths and up-down paths. A *directed path* in G, denoted in this work as $P(u_1, u_k)$, is a path that begins at u_1 and finishes at u_k where every edge

connecting the sequence of vertices is an outgoing edge of the previous element. An *up-down path* in G, denoted in this work as $P(u_1 : u_k)$, is a path that begins at u_1 and finishes at u_k where the edges connecting the sequence of vertices are incoming edges of the previous element up to an element u_p on the path such that $u_p \neq u_1$ and $u_p \neq u_k$ and all edges are outgoing edges of the previous element following u_p. The vertex u_p will be referred to as the *peak vertex* of the up-down path.

From the above definition, an (underlying) *cycle* of length k, where k is known a *k-cycle*, may be defined as a sequence of up-down paths $P(u_1 : u_a), ...P(u_b, u_c)$ where $u_1 = u_c$. This definition of a cycle is different to that of a directed cycle. A *directed cycle* is a directed path $P(u_1, u_k)$ where $u_1 = u_k$. Directed cycles are only relevant to this work in that they do not occur. From here on all underlying cycles will be simply referred to as cycles. This also allows a special type of edge to be characterised. If for two vertices u and v in a network there exists both a directed path $P(u, v)$ of length greater than two and and edge uv then the edge uv is called a *short-cut* or *redundant* edge.

Rooted binary phylogenetic networks. Throughout this work X will denote a finite non-empty set. This set is called a *leaf set* and each element a *leaf*. Though considered abstractly here, these sets represent identifiable data observed in the present. A *rooted binary phylogenetic network* $\mathcal{N}$ on X is a directed graph with no directed cycles and the following properties.

1. There exists precisely one vertex, the *root*, with in-degree zero and out-degree either zero or two.

2. The set of vertices in $\mathcal{N}$ with in-degree one and out-degree zero is X.

3. All other vertices have degree three.

The vertices with degree three in $\mathcal{N}$ will be referred to as *internal vertices*. In addition, the root and all vertices with in-degree one will be referred to as *tree vertices* and all vertices with in-degree more than one will be referred to as *reticulation vertices* or *reticulations*. This also

distinguishes the edges of the network into *reticulation edges*, incoming edges on reticulation vertices and *tree-edges*, all other edges.

Some definitions of phylogenetic networks allow the root of a network to have out-degree one. A root vertex of this type is called a *planted root*. The phylogenetic networks defined for this work do not have planted roots or internal vertices with degree-2. However, if $|X| = 1$ then $\mathcal{N}$ will be understood as a phylogenetic network that consists of a single vertex that is simultaneously the root and the only leaf. Finally, not directly following from the above properties but to remove unnecessary complexity 2-cycles, generally known as *parallel edges*, will also be not allowed. Where they occur, two edges both joining the same two vertices, one edge will simply be discarded from the edge set of the network and any resulting degree-2 edges suppressed. This will be referred to as *collapsing parallel edges*.

Phylogenetic X-trees. A *rooted binary phylogenetic X-tree* is a rooted binary phylogenetic network on X with no cycles or equivalently no reticulation vertices. Though historically a precursor to phylogenetic networks and often introduced first, phylogenetic trees exist as one of several subclasses of phylogenetic networks discussed in this work. Because of this, it is useful to view phylogenetic trees here by the structural restrictions that distinguish them from general phylogenetic networks.

With one exception the phylogenetic networks considered in this work are exclusively rooted binary phylogenetic networks. This exception occurs in Chapter 6 where phylogenetic networks are considered that have internal vertices with degree greater than three. In this chapter the definition of a phylogenetic network will be revisited. Otherwise, from here on rooted binary phylogenetic networks and rooted binary phylogenetic X-trees will be simply referred to as networks and phylogenetic X-trees respectively.

Several important features of a network can now be identified. Let $\mathcal{N}$ be a network on X and u and v two distinct vertices in $\mathcal{N}$. If there exists a directed path $P(u, v)$ from u to v, where $u \neq v$ then u is called an *ancestor* of v and v a *descendant* of u. For a vertex u the set of all leaves that descends from u is called the *cluster* set of u. A special case

of a network determined by a directed path is where u is the root v is a leaf and $P(u, v)$ contains every internal vertex of $\mathcal{N}$. In this case $\mathcal{N}$ is called a *caterpillar tree* and $P(u, v)$ the *spine* of the caterpillar. If there exists an up-down path $P(u : v)$ with peak vertex w then w is a *common ancestor* of both u and v. In the case where w is the parent vertex of both u and v the vertices u and v are known as *siblings*.

Cherries. A special case of sibling vertices is where the two vertices are leaves. The structure formed by these two leaves and their common parent vertex is called a *cherry*. If similarly there exists two leaves connected by an up-down path of length three containing precisely one reticulation-vertex the structure formed by this up-down path is called a *reticulated cherry*. Finally, if there exists two reticulated cherries that share the same reticulation-vertex then the two reticulated cherries are collectively called a *double reticulated cherry*. These structures are greatly exploited throughout the thesis to prove a number of results. Visual examples of these objects can be seen in Fig. 2.1.

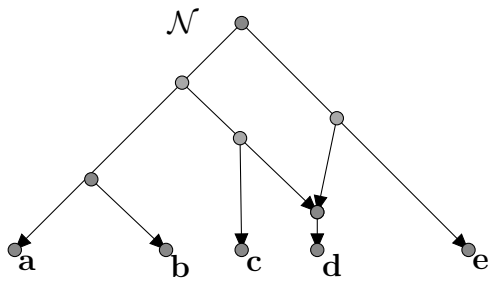

Figure 2.1: Cherries.
Here is a network $\mathcal{N}$ on leaf set $X = \{a, b, c, d, e\}$ where a and b are part of a cherry and both c and d and d and e are part of reticulated cherries and collectively a double reticulated cherry.

Deletion and restriction. In this work two main operations are applied to networks. The first is to delete an edge. Let $\mathcal{N}$ be a network on X with an edge uv. The *deletion* of uv, denoted $\mathcal{N} \setminus uv$, removes uv from the edge set of $\mathcal{N}$. This operation result in a graph that is not a network by the above definition. However, this is resolved in several particular ways.

Where the deletion of uv leaves resulting vertices with in-degree one or out-degree one, say u, the vertex u that has a parent vertex u' and

a child vertex u'' is also removed from the vertex set and its incoming and outgoing edges are replaced by a single edge $u'u''$ in the edge set. This is referred to as the *suppression of a degree-2 vertex*.

Where a vertex is left by the deletion of uv with either in-degree or out-degree zero, say v, the resulting network $\mathcal{N} \setminus uv$ will be taken to be the union of all paths from the root of $\mathcal{N}$ to a leaf in X that do not contain v with all resulting degree-2 vertices suppressed and parallel edges collapsed. In the case where this means there exists an element of X that is not a vertex in $\mathcal{N} \setminus uv$, the edge uv is called a *cut-edge* and the sets of vertices and edges that are not in $\mathcal{N} \setminus uv$ collectively make up a *pendant network* of $\mathcal{N}$. If v is a leaf then uv is a trivial cut-edge and this will be referred to as the *deletion of a leaf*, denoted $\mathcal{N} \setminus v$.

Finally, where the root of $\mathcal{N}$ is left by the deletion of uv with out-degree one, the root and its outgoing edge are removed from the vertex and edge sets of $\mathcal{N}$ and its child vertex is considered the root of $\mathcal{N} \setminus uv$. This is referred to as *removing a planted root*.

The second main operation applied to networks is to restrict the leaf set. Let $\mathcal{N}$ be a network on X and Y a subset of X. The *restriction* of $\mathcal{N}$ onto Y, denoted $\mathcal{N}|Y$ is the union of all paths from the root of $\mathcal{N}$ to a leaf in Y.

Displayed trees. The main interest of this work is the occurrence of one or multiple trees 'inside' a more complicated network. Let $\mathcal{N}$ be a network on X and $E_R(\mathcal{N})$ the set of reticulation edges in $\mathcal{N}$. The set $S \subset E_R(\mathcal{N})$ is a *switching set*, or simply a *switching*, in $\mathcal{N}$ if for every reticulation-vertex in $\mathcal{N}$ precisely one incoming reticulation edge is an element of S. For each switching S there is a unique complement switching $\bar{S} = \{E_R(\mathcal{N}) - S\}$. The switching S *yields* a phylogenetic X-tree $\mathcal{T}$ that is *displayed* by $\mathcal{N}$ if the deletion of every element of $\bar{S}$ produces $\mathcal{T}$. Note that in general the deletion of a complement switching may not yield an X-tree. In this work the set of phylogenetic X-trees displayed by a network $\mathcal{N}$ will be denoted $T(\mathcal{N})$. Where more than one switching in $\mathcal{N}$ yields the same tree that tree will be said to be displayed multiple times.

The trees displayed by a network will often be considered as they occur in the network. If S yields a phylogenetic X-tree $\mathcal{T}$ then the removal

of $\bar{S}$ from the edge set of $\mathcal{N}$, that allows any resulting planted roots, and degree-two vertices to remain, produces an *embedding* of $\mathcal{T}$ in $\mathcal{N}$. In this work this is denoted $\mathcal{N}_\mathcal{T}$ and will be referred to as an *embedded phylogenetic X-tree*. An example of this can be seen in Fig. 2.2.

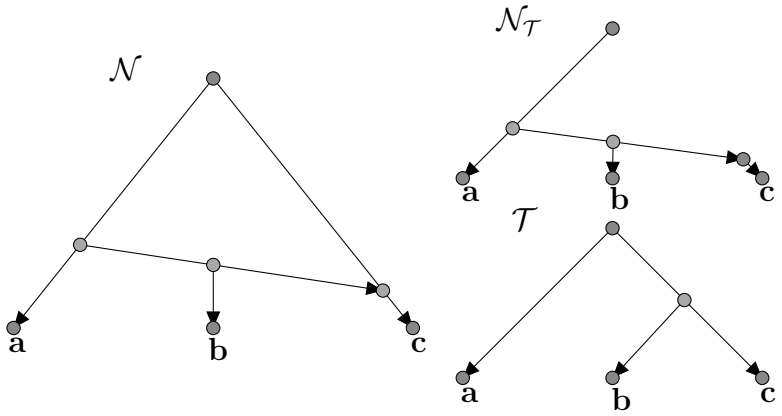

Figure 2.2: An embedded and displayed tree.
Here is a network $\mathcal{N}$ on leaf set $X = \{a, b, c\}$ that displays a phylogenetic X-tree $\mathcal{T}$ and contains the embedded phylogenetic X-tree $\mathcal{N}_\mathcal{T}$.

Tree-child networks. To explore the ways trees may exist inside and can be combined into networks this work considers several different classes of phylogenetic networks each distinguished by specific structural requirements. Central among these classes is the class of tree-child networks, introduced in [CRV09]. Let $\mathcal{N}$ be a network on X. If each non-leaf vertex in $\mathcal{N}$ has a tree vertex as a child then $\mathcal{N}$ is a *tree-child* network. Where two reticulation vertices share a parent vertex they will be called *sibling-reticulations* and where a reticulation-vertex is the parent of another the two reticulations will be called *stack-reticulations*. It can be observed that tree-child networks can also be characterised as the class of rooted binary phylogenetic networks that contain neither sibling nor stack-reticulations. This is proven in Proposition 2.0.1.

Proposition 2.0.1. *Let $\mathcal{N}$ be a network on X. The network $\mathcal{N}$ is tree-child if and only if $\mathcal{N}$ contains no sibling-reticulations or stack-reticulations. This is an independent proof of a result given in Theorem 1.1 of [Sem15].*

Proof. Let $\mathcal{N}$ be a network on X. Every non-leaf vertex in $\mathcal{N}$ is either

a tree vertex with two children or a reticulation-vertex with one. If a vertex v in $\mathcal{N}$ is a tree vertex with two reticulation vertices as children, sibling-reticulations, or a reticulation-vertex with a reticulation-vertex as its child, stack-reticulations, then v does not have a tree vertex as a child and $\mathcal{N}$ is not a tree-child network. If conversely there are no sibling or stack-reticulation vertices in $\mathcal{N}$ then every non-leaf vertex in $\mathcal{N}$ must have at least one tree vertex as a child. Hence $\mathcal{N}$ is tree-child. Therefore, a network $\mathcal{N}$ is tree-child if and only if $\mathcal{N}$ contains no sibling or stack-reticulations as required. □

Several key properties important to phylogenetic networks are vertices that have a tree-path to a leaf, vertices that are visible and networks that are tree-based. Let $\mathcal{N}$ be a network on X. A directed path in $\mathcal{N}$ is said to be a *tree-path* if every edge in the path is a tree edge. A vertex v is said to be *visible*, otherwise known as *stable*, in $\mathcal{N}$ if there exists a leaf in X such that every directed path from the root to that leaf in $\mathcal{N}$ contains v. Finally $\mathcal{N}$ is *tree-based* if there exists an embedded phylogenetic X-tree $\mathcal{N}_{\mathcal{T}}$ in $\mathcal{N}$ that contains every vertex in $\mathcal{N}$. This phylogenetic X-tree $\mathcal{T}$ is called a *base tree* of $\mathcal{N}$ and the class of tree-based networks is another important class of phylogenetic networks, introduced in [FS15], that is considered in this work.

Tree-child networks are particularly interesting because they have been found, separately in the literature, to exist as a meeting point between these properties. When the class of tree-based networks was introduced in [FS15] it was proven that every tree-child network is tree-based. This result was strengthened in [Sem15] where it was proven that the class of tree-child networks is precisely the class of networks for which every embedding of a tree displayed by the network is a base tree of the network. In addition to this tree-child networks can also be characterised in the following ways.

Proposition 2.0.2. *Let $\mathcal{N}$ be a network on X. The network $\mathcal{N}$ is tree-child if and only if for each vertex v in $\mathcal{N}$ there exists a tree-path from v to a leaf $\ell \in X$.*

Proof. Let $\mathcal{N}$ be a network on a leaf set X. If for each non-leaf vertex v in $\mathcal{N}$ there exists a tree-path to a leaf $\ell \in X$ then the next vertex in

the path must be a tree vertex. Hence each non-leaf vertex in $\mathcal{N}$ has a tree vertex as a child. Conversely, if $\mathcal{N}$ is tree-child then each non-leaf vertex v has a tree vertex as a child. A tree-path from any non-leave vertex to a leaf may then be found by taking a series of‘ subsequent tree vertex descendent until a leaf is found. Therefore a network $\mathcal{N}$ is tree-child if and only if for each vertex v in $\mathcal{N}$ there exists a tree-path from v to a leaf $\ell \in X$ as required. □

Proposition 2.0.3. *Let $\mathcal{N}$ be a network on X. The network $\mathcal{N}$ is tree-child if and only if every vertex v in $\mathcal{N}$ is visible.*

Proof. Let $\mathcal{N}$ be a network on X. Suppose that every vertex in $\mathcal{N}$ is visible but $\mathcal{N}$ is not tree-child. Because $\mathcal{N}$ is not tree-child there exists a vertex u that has no tree vertices as children. It follows from Proposition 2.0.1 that u is then either a tree vertex with two child reticulation vertices v_1 and v_2 or a reticulation-vertex with a single reticulation as a child, just v_1. Every child of u then has an additional parent vertex w_1 and w_2 or just w_1. In the case where one of w_1 or w_2 is a descendant of u, it must also be a descendant of the other additional parent vertex which is not a descendant of u. Let Y be the subset of X that descends from u. It is clear that every path from u to an element of Y must contain a child of u. As such for every leaf $\ell \in Y$ there exists a path from the root of $\mathcal{N}$ to ℓ that contains one of w_1 or w_2 and not u. Hence u is not a visible vertex in $\mathcal{N}$; a contradiction.

Conversely, suppose that $\mathcal{N}$ is tree-child and there exists a vertex u in $\mathcal{N}$ that is not visible. It follows from Proposition 2.0.2 that there exists a leaf $\ell_0 \in X$ such that the path $P(u, \ell_0)$ is a tree-path. As every element of $P(u, \ell_0)$ other then u necessarily has only one parent vertex there are no paths from the root to ℓ_0 that do not also contain u. Hence u is visible; a contradiction. Therefore a network $\mathcal{N}$ is tree-child if and only if every vertex v in $\mathcal{N}$ is visible as required. □

This chapter finishes with two final lemmas for tree-child networks. These lemmas, though not found explicitly are well-known in the literature. They will be used to identify a local structure that can be found in any tree-child network and what manipulations may be applied to that structure whilst preserving the tree-child property in the

resulting network. This then allows several inductive proofs to be made throughout this work.

Lemma 2.0.4. *Let $\mathcal{N}$ be a tree-child network on X where $|X| > 1$. Then $\mathcal{N}$ has either a cherry or a reticulated cherry. This is an independent proof of a result given in Lemma 4.1 of [BS15].*

Proof. Let $\mathcal{N}$ be a tree-child network on X, where $|X| > 1$, and with root ρ. Suppose $\mathcal{N}$ does not have either a cherry or a reticulated cherry.

Take a directed path from ρ to a leaf ℓ that is of maximum length in $\mathcal{N}$. Label this path $P(\rho, \ell)$. The leaf ℓ has a parent vertex v that is either a tree vertex or a reticulation-vertex. If v is a tree vertex then ℓ has a sibling vertex w. Because $\mathcal{N}$ has no cherries the sibling vertex w must have a child vertex. However, the directed path from ρ to this child vertex would be longer than $P(\rho, \ell)$; a contradiction.

If alternatively v is a reticulation-vertex then v has two parent vertices. Take the parent vertex that is also an element of $P(\rho, \ell)$. Label this parent vertex u. It follows from Proposition 2.0.1 that u is a tree vertex. As such v has a sibling vertex x. It again follows from Proposition 2.0.1 that x is a tree vertex. Because $\mathcal{N}$ has no reticulated cherries x must have two child vertices y and z. Because $\mathcal{N}$ has no cherries one of y or z must have a child vertex. However, if either of y or z has a child vertex then the directed path from ρ to this child vertex would be longer than $P(\rho, \ell)$; a contradiction. Therefore, all tree-child networks have either a cherry or a reticulated cherry as required. □

Lemma 2.0.5. *Let $\mathcal{N}$ be a tree-child network on X with a reticulation edge ur. The network $\mathcal{N} \setminus ur$ is tree-child.*

Proof. Let $\mathcal{N}$ be a tree-child network on X. Suppose that there exists a reticulation edge ur in $\mathcal{N}$ such that $\mathcal{N} \setminus ur$ is not tree-child.

The reticulation edge ur joins two vertices u and r where r is a reticulation. Because $\mathcal{N}$ is tree-child it follows from Proposition 2.0.1 that there can be no stack or sibling-reticulations in $\mathcal{N}$. From this it can be identified that u is a tree vertex that has another child vertex v that is also a tree vertex. Moreover, the same can be said for the other parent vertex of r, label this vertex w. As such, both parent vertices

of r have a tree vertex as a child in $\mathcal{N} \setminus ur$. Additionally ur is not a cut edge because every descendant of r remains a descendant of w in $\mathcal{N} \setminus ur$ insuring $\mathcal{N} \setminus ur$ is a network on X. Because every other vertex in $\mathcal{N}$ has the same child vertices in $\mathcal{N} \setminus ur$, it can then be seen that since every vertex in $\mathcal{N}$ has a tree vertex as a child so too does every vertex in $\mathcal{N} \setminus ur$. Hence $\mathcal{N} \setminus ur$ is a tree-child network; a contradiction. Therefore, for any tree-child network $\mathcal{N}$ with a reticulation edge ur the network $\mathcal{N} \setminus ur$ is tree-child as required. □

Chapter 3

Level-1 Network Construction

3.1 Introduction

In this chapter the network-capture problem is examined in the case of level-1 networks. This serves three purposes. First, in the restricted context of level-1 networks the underlying questions of this thesis can be elaborated on in detail. This gives a clear sign posting for the upcoming work on network classes in the subsequent chapters. Second, the differences that are found between level-1 and later networks will highlight the structural network properties that affect how and when trees are displayed by networks of different classes. Third, the results found for level-1 networks will be used to explore the underlying structure of more complex networks in later chapters.

Level-1 networks, otherwise known as galled trees, are a restricted generalisation of phylogenetic X-trees in which non-treelike events may occur but only in isolation. Because, like phylogenetic X-trees, level-1 networks are highly structured they are still very limited in what they can express. This makes them an ideal starting-point to explore the relationship between networks and the phylogenetic X-trees they display.

Formally introduced in [WZZ01] (2001) level-1 networks have been extensively studied. This is due in part to their simple structure and

because they occur as a subclass of many significant classes of network. The question of finding a level-1 network that displays a given set of trees has been explored following several different predominate models. These models are referred to as the'Tree-model' [BGMS05], 'Triplets-model' [VIKK$^+$08], 'Clusters model' and [HRB$^+$09] 'Binary-characters-model'[SWG05]. An overview and comparison of these different models is given in [VIK11]. Following from this several algorithms have in fact already been developed that reconstruct level-1 networks from given data, see [JS06], [HvIKS11], [OWVIM16]. However, instead of reconstructing the network directly from a set of phylogenetic X-trees (tree-model) these algorithms have all worked from sets of rooted triples (triplet-model).

A *rooted triple* is a phylogenetic X-tree where $|X| = 3$. It is well-known that any phylogenetic X-tree $\mathcal{T}$, where $|X| \geq 3$, can be characterised by the unique set of rooted triples comprised of the restrictions of $\mathcal{T}$ on each combination of three element subsets of X. This would intuitively suggest that constructing a network from the relevant set of rooted triples could be equivalent to constructing a network from a set of phylogenetic X-trees. The network-capture problem would then already be solved for level-1 networks. Despite this, two fundamental problems prevent these algorithms from being strictly suitable. These are that a set of rooted triples may not produce a network that displays the phylogenetic X-trees they characterise or reconstruct a known network they are obtained from.

The first and most significant problem was identified in [HvIKS11]. In Fig. 3.1 an example is given where $\mathcal{N}$, a phylogenetic X-tree and $\mathcal{N}'$, a network on X both share the all of the rooted triples in $\mathcal{N}$ but $\mathcal{N}'$ does not display $\mathcal{N}$. This means that for the purposes of constructing a network to display a given set of phylogenetic X-trees none of these algorithms working from rooted triples are appropriate. The next problem is explored in detail in [OWVIM16]. Here it is explained that there exists rooted triples that may be combined in different but equivalent ways. This is illustrated in Fig. 3.2 where despite being distinct networks, $\mathcal{N}$ and $\mathcal{N}'$ both display precisely the same set of phylogenetic X-trees $\mathcal{T}$ and $\mathcal{T}'$. This problem is addressed in [OWVIM16] by requiring a weight function for each rooted tree input. This solution

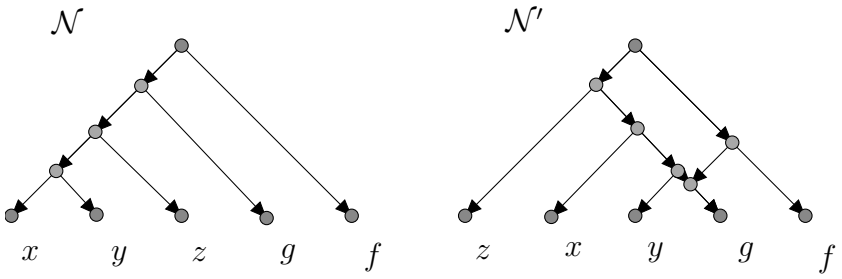

Figure 3.1: The network $\mathcal{N}'$ contains every rooted triple found in $\mathcal{N}$ however $\mathcal{N}'$ does not display $\mathcal{N}$. This example is from Figure 3 in [HvIKS11]

.

works well if there is additional data to build the weight function from. If however, the intention is to solely work from a set of phylogenetic X-trees again this is not appropriate.

In this chapter the second problem is addressed first. That there are different but equivalent ways of displaying some sets of rooted triples is not a problem when looking for a level-1 network to display a set of phylogenetic X-trees. This is because, as explained formally in Section 3.3, in a level-1 network each of these equivalent ways of displaying the same set of rooted triples are isolated from each other and can be interchanged since each option displays the same thing. This allows one of the multiple combinations to be chosen to be 'standard'. By choosing level-1 networks with all 4-cycles of the type of $\mathcal{N}$ in Fig. 3.2 to be 'standard' the first theorem of this chapter, Theorem 3.3.4, is proven. That is, a level-1 network of this chosen standard type $\mathcal{N}$ on X with tree display set $T(\mathcal{N})$ is the unique level-1 network of this chosen standard type that displays $T(\mathcal{N})$.

Next, in Section 3.4 the network-capture problem is solved for level-1 networks. This is done by producing a polynomial time algorithm, LEVEL-1 CONSTRUCT that takes a given set of phylogenetic X-trees as input and correctly answers if there exists a level-1 network that

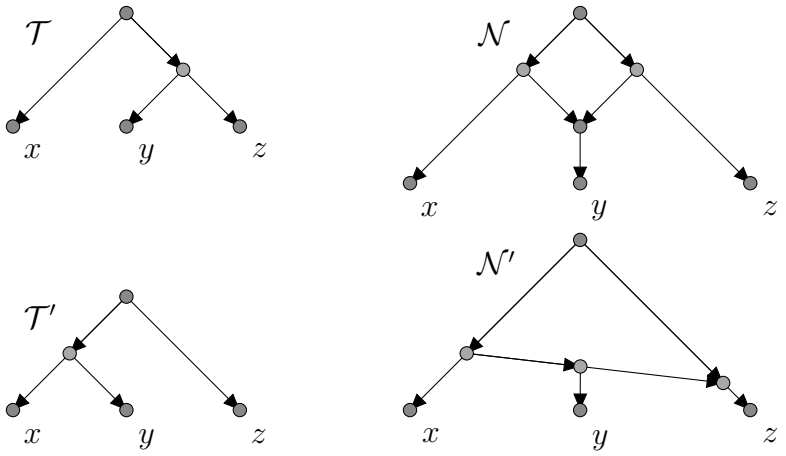

Figure 3.2: The two rooted triples $\mathcal{T}$ and $\mathcal{T}'$ are both displayed with minimal reticulations by both $\mathcal{N}$ and $\mathcal{N}'$. This example is derived from Figure 1 in [OWVIM16].

displays each given phylogenetic X-tree. Furthermore, in the positive case the algorithm outputs such a network.

In this chapter the network-capture problem is comprehensively answered for level-1 networks. However, an important consequence of the restricted structure of level-1 networks is that for many sets of trees there exists no level-1 network that displays them. This can be quickly identified by considering how fast the number of possible phylogenetic X-trees grows compared to the number of trees that a level-1 network on X can display grows as the number of elements in X increases. For $|X| = n$, it was identified in [Sch70] that the number of different phylogenetic X-trees is given by

$$\frac{(2n-2)!}{(n-1)!2^{n-1}}$$

In the next chapter it is conjectured that the maximum number of phylogenetic X-trees displayed by a level-1 network is less than or equal to $2^{(n-1)/2}$. These two equations swiftly diverge for $n > 2$, meaning many phylogenetic X-trees simply cannot be displayed together by a level-1 network.

This chapter is organised as follows. First, the formal definitions and

relevant known results necessary to begin are given in the preliminary section. In Section 3.3 the ambiguity caused by 3 and 4-cycles to phylogenetic X-trees displayed by level-1 networks is addressed by defining standard level-1 networks. From this the first theorem of this thesis is then proven. In Section 3.4, a polynomial time algorithm is produced that constructs, where possible, a level-1 network from a given set of phylogenetic X-trees. This resolves the network capture problem for level-1 networks. Then finally, the results of this chapter are summarised and discussed in Section 3.5.

3.2 Preliminaries

In this section, the definitions and known results that are necessary for this chapter are given. Specifically, level-1 networks are formally defined and their relationship to tree-child and tree-based networks is explained. This section then finishes with two lemmas that specify what affects deleting edges or restricting leaf sets have on level-1 networks.

Level-1 Networks. Let $\mathcal{N}$ be a network on X. If two cycles in $\mathcal{N}$ do not share any vertices then those two cycles are *vertex disjoint*. If every pair of cycles in $\mathcal{N}$ is vertex disjoint then $\mathcal{N}$ is a *level-1 network*.

Lemma 3.2.1. *Let $\mathcal{N}$ be a level-1 network on X. The network $\mathcal{N}$ is a tree-child network. (This result was first found as Lemma 3 in [CRV09] and is here given with an independent proof.)*

Proof. Let $\mathcal{N}$ be a level-1 network on X. Suppose that $\mathcal{N}$ is not tree-child. There then exists a vertex $v_0 \in \mathcal{N}$ that has no tree vertex child.

The vertex v_0 is either a tree vertex or a reticulation-vertex. If v_0 is a tree vertex then v_0 is the parent of two child reticulation vertices v_1 and v_2. There then exist two up-down paths $P(v_1 : v_1)$ and $P(v_2 : v_2)$ in $\mathcal{N}$ each characterising a separate cycle. However, it is clear that $v_0 \in P(v_1 : v_1)$ and $P(v_2 : v_2)$. As $\mathcal{N}$ is a level-1 network there exists no vertex that occurs in more than one cycle; a contradiction.

Alternatively if v_0 is a reticulation-vertex then v_0 is the parent of one child reticulation-vertex v_1. Similarly there then exist two up-down

paths $P(v_0 : v_0)$ and $P(v_1 : v_1) \in \mathcal{N}$ that each define a separate cycle. However again, $v_0 \in P(v_0 : v_0)$ and $P(v_1 : v_1)$; a contradiction. Therefore, $\mathcal{N}$ is tree-child as required. □

Lemma 3.2.2. *Let $\mathcal{N}$ be a level-1 network on X. The network $\mathcal{N}$ is a tree-based network.*

Proof. Let $\mathcal{N}$ be a level-1 network on X. It follows first from Lemma 3.2.1 that $\mathcal{N}$ is also a tree-child network and then from [Sem15] that $\mathcal{N}$ is tree-based as required. □

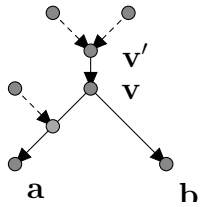

Figure 3.3: A reticulated cherry on leaves a and b.

From the next two lemmas the impact of deleting a reticulation edge or restricting the leaf set of a level-1 network is identified. The concern here is that these operations might produce a network that is not of the same class or displays a new set of trees that can not be understood in terms of specific alterations to the original set. In general this is very possible. For example consider a tree-child network $\mathcal{N}$ on X with a reticulated cherry of the type in Fig. 3.3. The vertex v is the peak vertex of the cherry and a child of a reticulation-vertex v'. In this case the deletion of the edge vb or the restriction of $\mathcal{N}|\{X - b\}$ will produce a network where v' has no tree vertex as a child. From this it can be seen that these operations when applied to tree-child networks may produce a network that is of a different class.

Lemma 3.2.3. *Let $\mathcal{N}$ be a level-1 network on X with a reticulation edge ur. The network $\mathcal{N} \setminus ur$ is also a level-1 network.*

Proof. Let $\mathcal{N}$ be a level-1 network on X with a reticulation edge ur. Because ur is a reticulation edge, r is a reticulation and it follows from Lemma 3.2.1 that u is a tree vertex. There then exists a path from the

root to each descendant of u and r that does not contain the edge ur. Hence $\mathcal{N} \setminus ur$ is a network on X.

Now suppose that $\mathcal{N} \setminus ur$ is not a level-1 network. If $\mathcal{N} \setminus ur$ is not a level-1 network there exists a vertex $v \in \mathcal{N} \setminus ur$ that is an element of at least two cycles C_1 and C_2 in $\mathcal{N} \setminus ur$. Let two such cycles be labelled C_1 and C_2. Every vertex in C_1 and C_2 is also a vertex of $\mathcal{N}$. If either u or r occurred as a child of a vertex in C_1 or C_2 in $\mathcal{N}$ then the vertex u or r would occur in more than one cycle in $\mathcal{N}$. However $\mathcal{N}$ is level-1 and every cycle in $\mathcal{N}$ is vertex disjoint; a contradiction. Hence the vertex v and the cycles C_1 and C_2 all occur in $\mathcal{N}$ as in $\mathcal{N} \setminus ur$. However again $\mathcal{N}$ is level-1 and every cycle in $\mathcal{N}$ is vertex disjoint; a contradiction. Therefore the deletion of a reticulation-vertex in a level-1 network produces a level-1 network as required. □

Lemma 3.2.4. *Let $\mathcal{N}$ be a level-1 network on X and $Y \subseteq X$. The restricted network $\mathcal{N}|Y$ is a level-1 network that displays each element of $T(\mathcal{N})$ restricted on to Y, the set $T(\mathcal{N})|Y$.*

Proof. Let $\mathcal{N}$ be a level-1 network on leaf set X and $T(\mathcal{N})$ the set of phylogenetic X-trees displayed by $\mathcal{N}$. If $\mathcal{N}|Y$ is not a level-1 network there exists a vertex v and two cycles C_1 and C_2 in $\mathcal{N}|Y$ such that v is an element of both C_1 and C_2. Each vertex in $\mathcal{N}|Y$ is a vertex $\mathcal{N}$. Moreover up to suppressed degree-two vertices every path in $\mathcal{N}|Y$ is a path in $\mathcal{N}$. This includes the up-down paths that define C_1 and C_2. Hence there exists a vertex common to two cycles in $\mathcal{N}$; a contradiction. Thus the restricted network $\mathcal{N}|Y$ is a level-1 network.

Next consider the set of phylogenetic X-trees $T(\mathcal{N})$ restricted onto Y, label this set $T(\mathcal{N})|Y$. Take one such restricted phylogenetic X-tree $\mathcal{T}|Y \in T(\mathcal{N}|Y)$. For $\mathcal{T}$ the corresponding non-restricted phylogenetic X-tree there exists an embedding $\mathcal{N}_\mathcal{T}$ in $\mathcal{N}$. It is clear that every vertex, edge and path in $\mathcal{N}_\mathcal{T}$ is a vertex, edge and path in $\mathcal{N}$. Additionally every vertex, edge and path in $\mathcal{N}_\mathcal{T}|Y$ is again a vertex, edge and path in $\mathcal{N}$. Just as the restriction of $\mathcal{T}$ on to Y gives $\mathcal{T}|Y$ so to does the restriction of $\mathcal{N}_\mathcal{T}$ on to Y if all degree-2 vertices are suppressed and all planted roots are removed. As such it follows that $\mathcal{N}|Y$ displays each tree in $\mathcal{T}|Y$. Therefore, The restricted network $\mathcal{N}|Y$ is a level-1 network that displays $T(\mathcal{N})|Y$ as required. □

3.3 Level-1 Uniqueness

In this section the results leading up to the main result of this section, Theorem 3.3.4, will be given. This begins with identifying what 3 and 4-cycles display. Here to distinguish between different 4-cycles, a 4-cycle that has the same number of vertices before and after the peak vertex of the up-down path that characterises them will be called a *symmetric 4-cycle*, for example $\mathcal{N}$ in Fig. 3.2 and those that do not will be called an *asymmetric 4-cycle*, for example $\mathcal{N}'$ in Fig. 3.2. Because all cycles are vertex disjoint in level-1 networks the way in which 3-cycles and 4-cycles display underlying tree-structure can be understood apart from the rest of the network.

For simplicity and clarity, interest is typically placed on networks that display a set of trees with as few reticulations as possible. This will be referred to in two distinct ways. Let $\mathcal{N}$ be a phylogenetic network on X that displays T, a set of phylogenetic X-trees. The network $\mathcal{N}$ displays T with *minimal* reticulations if the deletion of any reticulation edge in $\mathcal{N}$ would prevent the display of an element of T. On the other hand, the network $\mathcal{N}$ displays T with *minimum* reticulations if there exists no other network that displays T with a lesser number of reticulations. As will be demonstrated in the next chapter, generally a network that displays a set of trees with minimal reticulations may not display the set with minimum reticulations.

Lemma 3.3.1. *Let $\mathcal{N}$ be a tree-child network on X with a 3-cycle that is vertex disjoint from all other cycles. There exists a phylogenetic X-tree that is embedded in $\mathcal{N}$ in two different ways.*

Proof. Let $\mathcal{N}$ be a tree-child network on X with C a 3-cycle that is vertex disjoint from all other cycles. Take $\mathcal{T}$ to be a phylogenetic X-tree that is displayed by $\mathcal{N}$ via an embedded tree $\mathcal{N}_\mathcal{T}$.

As phylogenetic X-trees do not have any cycles $\mathcal{N}_\mathcal{T}$ contains only one of the two reticulation edges in C. Because every vertex in $\mathcal{N}$ is a vertex in $\mathcal{N}_\mathcal{T}$, replace this reticulation edge with the other one in $\mathcal{N}_\mathcal{T}$ to obtain an embedded tree $\mathcal{N}'_\mathcal{T}$. It can be quickly observed that as there are no cycles in $\mathcal{N}_\mathcal{T}$ there can be no cycles in $\mathcal{N}'_\mathcal{T}$ and there is

a directed path from the root to each element of X in $\mathcal{N}'_{\mathcal{T}}$. Hence by removing any planted roots and suppressing any internal degree-2 vertices in $\mathcal{N}'_{\mathcal{T}}$, a phylogenetic X-tree $\mathcal{T}'$ is obtained. Moreover, every path in $\mathcal{T}'$ is isomorphic to a path in $\mathcal{T}$. Specifically, every path is the same except for one pair where each has a different parent vertex of the reticulation in C. This means that both $\mathcal{N}_{\mathcal{T}}$ and $\mathcal{N}'_{\mathcal{T}}$ exist in $\mathcal{N}$ and display the same phylogenetic X-tree. Therefore, there exists a phylogenetic X-tree that is embedded in $\mathcal{N}$ in two different ways as required. □

Lemma 3.3.2. *Let $\mathcal{N}$ be a network on X with an asymmetric 4-cycle that is vertex disjoint from all other cycles. The network $\mathcal{N}'$ that is obtained by substituting the asymmetric 4-cycle in $\mathcal{N}$ with a symmetric four-cycle displays the same set of phylogenetic X-trees.*

Proof. Let $\mathcal{N}$ be a network on X with an asymmetric 4-cycle C that is vertex disjoint from all other cycles. There are precisely three vertices, a, b and c that are the child vertices of elements in C that are not themselves elements of C.

By separately deleting one reticulation edge and then the other in C two distinct sets of relationships are formed between a, b and c. Every embedded phylogenetic X-tree has one of these two relationships between a, b and c. Locally two of a, b and c occur in a cherry like relationship in one and another two in a cherry like relationship in the other. One of a, b and c will occur in both cherry like relationships, without loss of generality say this is b. Now construct $\mathcal{N}'$ by keeping the peak vertex of C, a, b and c fixed but replacing the asymmetric 4-cycle with a symmetric 4-cycle with a, b and c as children. Specifically make b the child of the reticulation-vertex and label this symmetric 4-cycle C'.

Now by again separately deleting one reticulation edge and then the other in C' two distinct sets of relationships are formed between a, b and c. Again locally two of a, b and c occur in a cherry like relationship in one and another two in a cherry like relationship in the other with b occurring in both. As every embedded phylogenetic X-tree has one of these two relationships between a, b and c and all other edges and

vertices remain the same it can be observed that every phylogenetic X-tree displayed by $\mathcal{N}$ is also displayed by $\mathcal{N}'$. Therefore, $\mathcal{N}$ and $\mathcal{N}'$ both displays the same set of phylogenetic X-trees as required. □

Next it is proven that for level-1 networks a network on X that displays a set of phylogenetic X-trees with minimal reticulations also displays that set with minimum reticulations. However, as will be proven in Proposition 4.3.6 not all classes of phylogenetic networks share this property.

Lemma 3.3.3. *Let $\mathcal{N}$ be a level-1 network on X that displays a set of phylogenetic X-trees T with a minimal number of reticulations. The network $\mathcal{N}$ is a level-1 network that displays T with the minimum reticulations.*

Proof. Let $\mathcal{N}$ be a level-1 network on X that displays a set of phylogenetic X-trees T with a minimal number of reticulations m. Suppose that there exists another level-1 network $\mathcal{M}$ on X that displays T with minimal reticulations and the number of these reticulations is less than m.

It follows from Lemma 3.2.1 and Lemma 3.2.2 that, as level-1 networks $\mathcal{N}$ and $\mathcal{M}$, both are tree-child and tree-based networks. From [Sem15] it then follows that every phylogenetic X-tree in T is a base tree of both $\mathcal{N}$ and $\mathcal{M}$. Take a phylogenetic X-tree $\mathcal{T} \in T$. For two embeddings of $\mathcal{T}$, $\mathcal{N}_\mathcal{T}$ and $\mathcal{M}_\mathcal{T}$, let the two sets of edges $E_\mathcal{N}$ in $\mathcal{N}$ and $E_\mathcal{M}$ in $\mathcal{M}$ be the edges of $\mathcal{N}$ that are not in $\mathcal{N}_\mathcal{T}$ and $\mathcal{M}$ that are not in $\mathcal{M}_\mathcal{T}$, respectively.

Since $\mathcal{T}$ is a base tree each element in $E_\mathcal{N}$ or $E_\mathcal{M}$ produces a cycle when added to $\mathcal{N}_\mathcal{T}$ and $\mathcal{M}_\mathcal{T}$ respectively. It follows from Lemma 3.3.1 that any 3-cycle in $\mathcal{N}$ and $\mathcal{M}$ can be replaced with a local cherry like structure and T would still be displayed. Therefore it may be assumed that every cycle in $\mathcal{N}$ and $\mathcal{M}$ has at least four vertices.

For each cycle in $\mathcal{N}$ it follows from Lemma 3.2.1 and Proposition 2.0.2 that for every vertex in a cycle, a descendant leaf may be chosen that is the descendant of no other vertex in the same cycle. Hence for each cycle C in $\mathcal{N}$, a set of leaves $Y_C \subset X$ may be chosen so that $\mathcal{N}|Y_C$ is

a level-1 network (possibly with a planted root) containing the cycle C and no other cycle. Every restricted network $\mathcal{N}|Y_C$ displays precisely two distinct phylogenetic Y_c-trees, $\mathcal{T}|Y_C$ and $\mathcal{T}'|Y_C$. Because $\mathcal{N}$ displays the elements of T with minimal reticulations it follows from Lemma 3.2.4 that each $\mathcal{N}|Y_C$ displays $T|Y_C$. The network $\mathcal{M}$ also displays every element of T with minimal reticulations. Hence, it follows again from Lemma 3.2.4 that for each Y_C, the restricted network $\mathcal{M}|Y_C$ also displays $\mathcal{T}|Y_C$ and $\mathcal{T}'|Y_C$. However, as there are no 3-cycles in either network in no case does $\mathcal{T}|Y_C = \mathcal{T}'|Y_C$. This means each $\mathcal{M}|Y_C$ has at least 1 reticulation-vertex and because there are m cycles C there exists m reticulation vertices in $\mathcal{M}$; a contradiction. Therefore, $\mathcal{N}$ is a level-1 network that displays T with the minimum number of reticulation vertices as required. □

Because all cycles in a level-1 network are vertex disjoint the ambiguity around four-cycles may be accounted for by simply choosing to use only symmetric four-cycles. This choice may be made as it follows from Lemma 3.3.2 that the replacement of a asymmetric 4-cycle with a symmetric 4-cycle in a level-1 network does not change what phylogenetic X-trees are displayed. Here level-1 networks with all four-cycles replaced as symmetric four-cycles will be referred to as *standard level-1 networks*. This all then leads to Theorem 3.3.4 which will be used to resolve the network capture problem in the next section an underpin the later work in Chapter 5.

Theorem 3.3.4. *Let T be a set of phylogenetic X-trees on a common leaf set X. If there exists a standard level-1 network $\mathcal{N}$ that displays each element of T with minimum reticulations then $\mathcal{N}$ is the unique standard level-1 network that displays T with minimum reticulations.*

Proof. Let T be a set of phylogenetic X-trees on a common leaf set X and $\mathcal{N}$ a level-1 network on X that displays each element of T with minimal reticulations. The following is a proof by induction on $|X|$. For $|X| = 1$ the set T is comprised of one phylogenetic X-tree $\mathcal{T}_1$ where $\mathcal{T}_1$ is a single vertex. It is then clear that the standard level-1 phylogenetic network $\mathcal{N}$, which is also a single vertex, uniquely displays

$\mathcal{T}_1$ as required. Assume that for $|X| = k$, the network $\mathcal{N}$ is the unique standard level-1 network that displays T.

Now let $|X| = k + 1$. It follows from Lemma 3.2.1 and Lemma 2.0.4 that $\mathcal{N}$ has either a cherry or reticulated cherry. If there exists a cherry $\{\ell_1, \ell_2\}$ in $\mathcal{N}$, delete the leaf ℓ_1 from X to construct the leaf subset $Y \subset X$. It follows from Lemma 3.2.4 that $\mathcal{N}|Y$ is a level-1 network. Moreover, because $\{\ell_1, \ell_2\}$ is a cherry in $\mathcal{N}$ and $\mathcal{N}$ displays every element of T with minimal reticulations $\mathcal{N}|Y$ also displays each element in $T|Y$ with minimal reticulations. Hence, by the above assumption, $\mathcal{N}|Y$ uniquely displays each element in $T|Y$ with minimal reticulations. Because $\{\ell_1, \ell_2\}$ is a cherry in $\mathcal{N}$, the cherry $\{\ell_1, \ell_2\}$ must occur in each element of T. Thus to produce a level-1 network that displays T by connecting ℓ_1 to $\mathcal{N}|Y$ the leaf ℓ_1 can only be connected via the incoming edge on ℓ_2 as in $\mathcal{N}$. This means that $\mathcal{N}$ is the unique level-1 network that displays T with minimal reticulations.

If, on the other hand, there are no cherries in $\mathcal{N}$ then there is a reticulated cherry $\{\ell_1, \ell_2\}$ with reticulation-vertex r. Without loss of generality let ℓ_1 be the child vertex of r in the reticulated cherry. The up-down path of minimal length starting and finishing at r with peak vertex v then describes a cycle C in $\mathcal{N}$. Because $\mathcal{N}$ is a standard level-1 network that displays the trees in T with minimal reticulations C must not be a 3-cycle and hence is a symmetric 4-cycle or a cycle on more than four vertices. Now delete the leaf ℓ_1 from X to construct the leaf subset $Y \subset X$. It follows from Lemma 3.2.4 that $\mathcal{N}|Y$ is a level-1 network that displays $T|Y$. Again because $\mathcal{N}$ displays T with minimal reticulations it can be seen that $\mathcal{N}|Y$ also displays $T|Y$ with minimal reticulations. Hence, by the above assumption, $\mathcal{N}|Y$ uniquely displays $T|Y$ with minimal reticulations.

Now to complete this proof it will be shown that there is again only one way to produce a level-1 network that displays T with minimal reticulations by connecting ℓ_1 to $\mathcal{N}|Y$. Consider just C in $\mathcal{N}$. Locally, v may be considered as the root and the child vertices of elements of C that are not also elements of C as leaves. In treating C as a level-1 network inside $\mathcal{N}$ it can be seen that C can only display two local tree-structures. These pairs of local tree-like structures can be of two

types. Either there is a single edge from v to r or there is another parent vertex of r, other than the parent of ℓ_2. Label this other vertex u and consider the two cases where either vr or ur is an edge in $\mathcal{N}$.

In the former case, where there is a single edge comprising the up or the down portion of the up-down path defining C, locally any tree displayed by $\mathcal{N}$ has $\{\ell_1, \ell_2\}$ as a cherry or ℓ_1 and ℓ_2 at extreme ends of a local caterpillar tree-structure. Because $\mathcal{N}$ displays T with minimal reticulations there exists at least two trees in T where one has the first local structure and the other the second. Hence, to produce a level-1 network that displays T with minimal reticulations by connecting ℓ_1 to $\mathcal{N}|Y$ a directed path to ℓ_1 must join $\mathcal{N}|Y$ from the incoming edge to ℓ_2 and another must join $\mathcal{N}|Y$ from the outgoing edge from v. These two paths can not be the same and thus must join at a single reticulation-vertex to preserve minimality. This would be the same as in $\mathcal{N}$.

In the latter case, the vertex u is a parent vertex of r and another vertex u' in $\mathcal{N}$. Here locally, u' is treated as a leaf and any tree displayed by $\mathcal{N}$ has $\{\ell_1, \ell_2\}$ as a cherry or ℓ_1, u' as siblings. Again, because $\mathcal{N}$ displays T with minimal reticulations there exists at least two trees in T where one has the first local structure and the other the second. Hence, to produce a level-1 network that displays T with minimal reticulations by connecting ℓ_1 to $\mathcal{N}|Y$ a directed path to ℓ_1 must join $\mathcal{N}|Y$ from the incoming edge to ℓ_2 and another must join $\mathcal{N}|Y$ from the incoming edge to u'. These two paths can not be the same and thus must join at a single reticulation-vertex to preserve minimality. This would be the same as in $\mathcal{N}$. Thus $\mathcal{N}$ is the unique standard level-1 network that displays T with minimal reticulations. Therefore, by induction if there exists a standard level-1 network $\mathcal{N}$ that displays a set of phylogenetic X-trees with minimal reticulations then $\mathcal{N}$ is the unique such network as required. □

3.4 Level-1 Construction

In this section the network-capture problem is resolved for level-1 networks. Here a polynomial time algorithm, LEVEL-1 CONSTRUCT is produced that identifies if, for a given set T of phylogenetic X-trees there

exists a level-1 network on X that displays each tree in T. Furthermore, in the positive case the algorithm LEVEL-1 CONSTRUCT outputs the unique standard level-1 network $\mathcal{N}$ on X that displays each tree in T with minimum reticulations.

The algorithm LEVEL-1 CONSTRUCT works by exploiting the defining property of level-1 networks, that every cycle in a level-1 network is vertex disjoint. With every vertex in a level-1 network being vertex disjoint every cycle has precisely one reticulation and displays two local tree structures. It follows from Lemma 3.3.1 that in the case of 3-cycles the same local tree structure is displayed twice. However, by only considering sets of trees rather than multi-sets of trees, sets where an element may occur more than once, 3-cycles need not be taken into account. Without 3-cycles, it can be quickly identified that only a small number of specific pairs of local tree-structures can be displayed by a cycle with one reticulation. By identifying pendant trees in elements of T that comprise either a cherry or a cycle from which no other cycle descends in any possible level-1 network that displays T the algorithm LEVEL-1 CONSTRUCT either takes apart the elements of T one pendant tree at a time and reconstructs $\mathcal{N}$ or identifies that there exists no level-1 network that displays the elements of T.

Cherry triple. To begin, the relevant pendants in a given set of trees must be identified. Let T be a set of phylogenetic X-trees where $|X| \geq 1$. If there exists a level-1 network on X that displays each tree in T, then the set T will be said to be *level-1 compatible*. For leaves a, b and c in X, where b and c may not be distinct, a *cherry triple* $\{a, (b, c)\}$ of T is a set of three leaves such that every tree in T has either the cherry $\{a, b\}$ or $\{a, c\}$. Moreover, when b and c are distinct, there exists at least one tree in T with each cherry.

Lemma 3.4.1. *Let T be a set of level-1 compatible phylogenetic X-trees where $|X| > 1$. There exists a cherry triple in T.*

Proof. Let T be a set of level-1 compatible phylogenetic X-trees where $|X| > 1$. It follows from Theorem 3.3.4 that there exists a unique standard level-1 network $\mathcal{N}$ on X that displays each tree in T with minimum reticulations. In the following cases, the leaves a, b and c

in X, where b and c may not be distinct, will be used repetitively to give an immediate connection to the labelling of $\{a,(b,c)\}$ as a cherry triple.

If there is a cherry $\{a,b\}$ in $\mathcal{N}$ then every tree displayed by $\mathcal{N}$ has the cherry $\{a,b\}$. Hence, where $b=c$ there is a cherry triple $\{a,(b,c)\}$ in T. Alternatively, if there are no cherries in $\mathcal{N}$ because X is finite there exists a pendant level-1 network $\mathcal{P}$ in $\mathcal{N}$ where the root of $\mathcal{P}$ and every internal vertex is an element of a single cycle. Label the leaf set of $\mathcal{P}$ as Y, where $Y \subseteq X$ and a, b and c will now be elements of Y.

Because there is one cycle in $\mathcal{P}$ there is precisely one reticulation-vertex r with two incoming reticulation edges. As such, $\mathcal{P}$ displays two Y-trees or one Y-tree twice in the case that the cycle in $\mathcal{P}$ is a 3-cycle. Where there is a 3-cycle in $\mathcal{P}$ the leaf set Y consists of a and b. Just as before, every tree displayed by $\mathcal{N}$ then has the cherry $\{a,b\}$. Hence, where $b=c$ there is a cherry triple $\{a,(b,c)\}$ in T. Otherwise, if there is a short-cut in $\mathcal{P}$, then r has a child which is a leaf c and a sibling which is also a leaf a. Similarly, the parent vertex of a has a sibling which is a leaf b. From this it can be observed that one of the Y-trees displayed by $\mathcal{P}$ is a caterpillar tree with the cherry $\{a,b\}$ and the leaf c as the child of the root and the other is a caterpillar tree with the cherry $\{a,c\}$. If there is no short-cut in $\mathcal{P}$ then r has a child that is a leaf a and two siblings that are leaves b and c. It can again be observed that one of the Y-trees displayed by $\mathcal{P}$ has the cherry $\{a,b\}$ and the other has the cherry $\{a,c\}$. Moreover, each phylogenetic X-tree displayed by $\mathcal{N}$ has one of the trees in these two pairs of Y-trees as a pendant tree. Thus finally, $\{a,(b,c)\}$ is a cherry triple in T as required. □

After phylogenetic X-trees, the most simple level-1 networks are those with at most one cycle. This special subclass of level-1 networks is known as *uni-cyclic networks*. It may be noted that with at most one cycle and consequently one reticulation a uni-cyclic network on X can display at most two phylogenetic X-trees. If for a set T of phylogenetic X-trees there exists a uni-cyclic network on X that displays each tree in T then the set T will be called *uni-cyclic compatible*. Where $|T|=1$ the set T will be considered trivially uni-cyclic compatible.

Level-1 reduction. This next operation will be used to reduce a set

of phylogenetic X-trees by one cherry or an appropriate pendant tree one at a time. Let T be a level-1 compatible set of phylogenetic X-trees where $|X| > 1$. For a cherry triple $\{a, (b, c)\}$ in T let $C_{\{a,(b,c)\}}$ denote the minimal cluster that contains each leaf in $\{a, (b, c)\}$ for all trees in T. Call $C_{\{a,(b,c)\}}$ the *cherry cluster* of $\{a, (b, c)\}$ in T. Among all cherry triples in T let $C^T_{\{a,(b,c)\}}$ denote a cherry cluster of minimal size in T. Call $C^T_{\{a,(b,c)\}}$ a *pendant cluster* of T. For a pendant cluster $C^T_{\{a,(b,c)\}}$ in T the *level-1 reduction* of T via $C^T_{\{a,(b,c)\}}$ is the restriction of each tree in T on to the leaf set $X' = \{X - \{C^T_{\{a,(b,c)\}} - b\}\}$ giving a new set T' of phylogenetic X'-trees.

Lemma 3.4.2. *Let T be a set of uni-cyclic compatible phylogenetic X-trees, where $|X| \geq 3$. For the restriction of the trees in T onto a pendant cluster $C^T_{\{a,(b,c)\}}$ in T one of the following is true.*

1. *The leaves $b = c$ and both trees in T share a cherry $\{a, b\}$.*
2. *The leaves $b \neq c$, both trees in T are caterpillar trees and one of b or c occurs in one tree as an element of the cherry and in the other tree as a child of the root. In this case the only uni-cyclic network that displays the trees in T has a short-cut and the above described leaf as the child of the reticulation.*
3. *The leaves $b \neq c$ and at most one tree in T is a caterpillar tree. In this case the only uni-cyclic network that displays the trees in T has no short-cuts and has a as the child of the reticulation.*

Proof. Let T be a set of uni-cyclic compatible phylogenetic X-trees, where $|X| \geq 4$. As T is uni-cyclic compatible and every cycle in a uni-cyclic network is trivially vertex disjoint, T is also level-1 compatible. It follows from Lemma 3.4.1 that there exists a pendant cluster $C^T_{\{a,(b,c)\}}$ in T. From Theorem 3.3.4, Lemma 2.0.4 and Lemma 3.2.1 it also follows that there exists a unique standard level-1 network $\mathcal{N}$ that displays the trees in T and there exists either a cherry or a reticulated cherry in $\mathcal{N}$.

If there exists a cherry in $\mathcal{N}$ the leaves $b = c$ and every tree displayed by $\mathcal{N}$ also shares the cherry $\{a, b\}$. Otherwise, there are no cherries in $\mathcal{N}$ and every non-leaf vertex in $\mathcal{N}$ is the element of a single cycle C. Either there exists a short-cut in C or there does not. If there is a short-cut in

$\mathcal{N}$ both trees displayed by $\mathcal{N}$ are caterpillar trees and the one of b or c that descends from the reticulation occurs in one tree as an element of the cherry and in the other tree as the child of the root. If there is no short-cut in $\mathcal{N}$ then a descends from the reticulation and at most one of the trees displayed by $\mathcal{N}$ is a caterpillar tree as required. □

Lemma 3.4.3. *Let T be a set of phylogenetic X-trees and T' the resulting set after repetitively taking the level-1 reduction of T until there are no common cherries among the elements of T. The set T is uni-cyclic compatible if and only if either T' consists of a single element which is a single vertex or there exists precisely one pendant cluster $C^{T'}_{\{a,(b,c)\}}$ in T' and the elements of T' are as described by option 2. or 3. from Lemma 3.4.2.*

Proof. For the first direction of this proof, let T be a set of phylogenetic X-trees and T' the resulting set after repetitively taking the level-1 reduction of T until there are no common cherries among the elements of T. It follows directly from Lemma 3.4.2 that if T is uni-cyclic compatible then either T' consists of a single element which is a single vertex or there exists precisely one pendant cluster $C^{T}_{\{a,(b,c)\}}$ in T' and the elements of T' are as described by option 2. or 3. from Lemma 3.4.2.

For the second direction of this proof, let T be a set of phylogenetic X-trees and T' the resulting set after repetitively taking the level-1 reduction of T until there are no common cherries among the elements of T. Suppose that either T' consists of a single element which is a single vertex or there exists precisely one pendant cluster $C^{T}_{\{a,(b,c)\}}$ in T' and the elements of T' are as described by option 2. or 3. from Lemma 3.4.2.

Consider first the case that T' consists of a single element which is a single vertex. This means every cherry in an element of T is a cherry in every other element of T and this remains true after the removal of each such cherry. As such T consists of a single phylogenetic X-tree and T is uni-cyclic compatible.

Consider second the case that there exists precisely one pendant cluster $C^{T}_{\{a,(b,c)\}}$ in T' and the elements of T' are as described by option 2. from

Lemma 3.4.2. The trees in T' are then caterpillar trees were one of b or c occurs as an element of the cherry in some elements T' and the same leaf, either b or c, occurs as the child of the root in the other elements of T'. Without loss of generality let b be the leaf as described as above and $\mathcal{T}$ a tree in T' where b is not an element of a cherry. By joining an edge from the incoming edge on a to the incoming edge on b in $\mathcal{T}$ a uni-cyclic network is produced that displays all elements of T'. By adding in reverse order all cherries removed from T to this uni-cyclic network a uni-cyclic network is produced that displays all elements of T. Hence T is uni-cyclic compatible.

Consider third the case that there exists precisely one pendant cluster, $C^T_{\{a,(b,c)\}}$ in T' and the elements of T' are as described by option 3. from Lemma 3.4.2. Label a tree in T' in which $\{a, b\}$ is a cherry $\mathcal{T}$. By joining an edge from the incoming edge on c to the incoming edge on a in $\mathcal{T}$ a uni-cyclic network is produced that displays all elements of T'. By adding in reverse order all cherries removed from T to this uni-cyclic network a uni-cyclic network is produced that displays all elements of T. Hence T is uni-cyclic compatible.

Therefore, a set of phylogenetic X-trees is uni-cyclic compatible if and only if after removing all common cherries the following is true. The resulting set either consists of a single element that is a single vertex or there exists precisely one pendant cluster $C^{T'}_{\{a,(b,c)\}}$ in T' and the elements of T' are as described by option 2. or 3. from Lemma 3.4.2 as required. □

Lemma 3.4.4. *Let T be a set of level-1 compatible phylogenetic X-trees where $|X| > 1$. For a pendant cluster $C^T_{\{a,(b,c)\}}$ in T the following is true.*

1. *The restriction of each tree in T onto $C^T_{\{a,(b,c)\}}$ gives a set $T|C^T_{\{a,(b,c)\}}$ of $C^T_{\{a,(b,c)\}}$-trees where each $C^T_{\{a,(b,c)\}}$-tree is a pendant tree of a phylogenetic X-tree in T. Moreover, the number of elements in $T|C^T_{\{a,(b,c)\}}$ is less than or equal to 2 and $T|C^T_{\{a,(b,c)\}}$ is uni-cyclic compatible.*

2. *For a level-1 network $\mathcal{N}$ on X that displays each element of T, the restriction of $\mathcal{N}$ onto $C^T_{\{a,(b,c)\}}$ gives a standard level-1 net-*

work $\mathcal{N}|C^T_{\{a,(b,c)\}}$ *that displays every element of* $T|C^T_{\{a,(b,c)\}}$ *and is a pendant network of* $\mathcal{N}$*.*

3. *Either every element of* $\mathcal{N}|C^T_{\{a,(b,c)\}}$ *is an element of a cherry or the root and every internal vertex of* $\mathcal{N}|C^T_{\{a,(b,c)\}}$ *is an element of a single cycle.*

Proof. Let T be a set of level-1 compatible phylogenetic X-trees where $|X| > 1$. It follows from Lemma 3.4.1 that there exists a pendant cluster $C^T_{\{a,(b,c)\}}$ in T. Additionally, as T is level-1 compatible it also follows from Theorem 3.3.4 that there exists a unique standard level-1 network $\mathcal{N}$ on X that displays T with minimum reticulations. Now from Lemma 3.2.4 it is clear that the restricted network $\mathcal{N}|C^T_{\{a,(b,c)\}}$ is a level-1 network that displays every tree in the set of restricted trees $T|C^T_{\{a,(b,c)\}}$. Moreover, as $C^T_{\{a,(b,c)\}}$ is a cluster of each tree in T the leaf subset $C^T_{\{a,(b,c)\}}$ is a cluster of $\mathcal{N}$ making $\mathcal{N}|C^T_{\{a,(b,c)\}}$ a pendant network of $\mathcal{N}$.

If $b = c$ then every tree in T has the cherry $\{a, b\}$ and consequently so does $\mathcal{N}$. As such $\mathcal{N}|C^T_{\{a,(b,c)\}}$ consists of a root and both leaves in a cherry. Alternatively, as $C^T_{\{a,(b,c)\}}$ is a cherry cluster of minimal size in T if $b \neq c$ then there are no cherries in $\mathcal{N}|C^T_{\{a,(b,c)\}}$. Because every cycle in a level-1 network is vertex disjoint there exists precisely one reticulation in each cycle in $\mathcal{N}|C^T_{\{a,(b,c)\}}$. With no cherries, $\mathcal{N}|C^T_{\{a,(b,c)\}}$ has a cycle C from which no other cycles descend. The pendant network of $\mathcal{N}|C^T_{\{a,(b,c)\}}$ consisting of C and the leaves that descend from it, $\mathcal{P}$ is then a uni-cyclic network with no cherries. As such it follows from Lemma 3.4.2 that first if $\mathcal{P} \neq \mathcal{N}|C^T_{\{a,(b,c)\}}$ there exists a cherry cluster in T with less elements than $C^T_{\{a,(b,c)\}}$; a contradiction. Second, every non-leaf vertex in $\mathcal{N}|C^T_{\{a,(b,c)\}}$ is an element of a single cycle, $T(\mathcal{N}|C^T_{\{a,(b,c)\}})$ is uni-cyclic compatible and the number of elements in $T(\mathcal{N}|C^T_{\{a,(b,c)\}})$ is less than or equal to 2 as required. □

Lemma 3.4.5. *Let* T *be a set of phylogenetic* X*-trees with a pendant cluster* $C^T_{\{a,(b,c)\}}$ *and* T' *the set of phylogenetic* X'*-trees obtained by taking the level-1 reduction of* T *via* $C^T_{\{a,(b,c)\}}$*. The set* T *is level-1 compatible if and only if* T' *is level-1 compatible and* $T|C^T_{\{a,(b,c)\}}$ *is uni-cyclic compatible.*

Proof. Let T be a set of phylogenetic X-trees where $|X| > 1$. It follows from Lemma 3.4.1 that there exists a pendant cluster $C^T_{\{a,(b,c)\}}$ in T. Take T' to be the set of phylogenetic X'-trees obtained by taking the level-1 reduction of T via $C^T_{\{a,(b,c)\}}$. The following is a proof by induction on $|X|$.

For $|X| = 2$ the set of trees T consists of a single element, a root connected to two leaves in a cherry. The set T' also consists of a single element, a single vertex. With both sets consisting of single elements it is clear that T is level-1 compatible if and only if T' is level-1 compatible. Assume that for $|X| = k$, the set T is level-1 compatible if and only if T' is level-1 compatible and $C^T_{\{a,(b,c)\}}$ is uni-cyclic compatible. Now consider the case where $|X| = k + 1$.

For the first direction of this proof, it follows first from Theorem 3.3.4 that there exists a unique standard level-1 network $\mathcal{N}$ that displays T with minimum reticulations. It then follows second from Lemma 3.4.4 that the restricted network $\mathcal{N}|C^T_{\{a,(b,c)\}}$ is a pendant level-1 network of $\mathcal{N}$ that displays the set of restricted trees $T|C^T_{\{a,(b,c)\}}$, each element of which occur as a pendant tree of an element of T. As such, the replacement of the pendant network $\mathcal{N}|C^T_{\{a,(b,c)\}}$ in $\mathcal{N}$ with the single leaf b gives a level-1 network that displays each element of T'. Thus if T is level-1 compatible so to is T'.

For the second direction of this proof, let T' be level-1 compatible and $T|C^T_{\{a,(b,c)\}}$ uni-cyclic compatible. It follows again from Theorem 3.3.4 that there exists two unique standard level-1 networks, $\mathcal{N}'$ that displays T' with minimum reticulations and $\mathcal{N}|C^T_{\{a,(b,c)\}}$ that displays $T|C^T_{\{a,(b,c)\}}$ with minimum reticulations. It follows from Lemma 3.4.4 that every element of T has one of the at most two elements of $T|C^T_{\{a,(b,c)\}}$ as a pendant tree. From Lemma 3.4.2 it then follows that there exists a uni-cyclic network $\mathcal{P}$ that is constructible from observing the occurrence of leaves in each element of $T|C^T_{\{a,(b,c)\}}$. The replacement of the leaf b in $\mathcal{N}'$ with the $\mathcal{P}$ as a pendant network then produces a level-1 network that displays every tree in T. Thus T is level-1 compatible as required.

Therefore, by induction on $|X|$, the set T is level-1 compatible if and only if T' is level-1 compatible and $T|C^T_{\{a,(b,c)\}}$ is uni-cyclic compatible as required. □

Algorithm: LEVEL-1 CONSTRUCT

Input: A non-empty finite set X and a set T of phylogenetic X-trees.
Output: The unique standard level-1 network $\mathcal{N}$ on X that displays each tree in T with minimum reticulations or the statement 'T is not level-1 consistent'.

1. If $|X| = 1$, return the phylogenetic network consisting of a single vertex as $\mathcal{N}$.

2. If $|X| = 2$, return the phylogenetic network consisting of the two leaves in X joined to the root in a cherry as $\mathcal{N}$.

3. Else, find a pendant cluster $C^T_{\{a,(b,c)\}}$ in T;

   I If $C^T_{\{a,(b,c)\}} = X$ return the statement 'T is not level-1 consistent'.

   II Else, take the level-1 reduction T' of T via $C^T_{\{a,(b,c)\}}$. This gives a new set of phylogenetic X'-trees T' where $X' = \{X - \{C^T_{\{a,(b,c)\}} - b\}\}$. Apply LEVEL-1 CONSTRUCT to X' and T'. Construct $\mathcal{N}$ from the returned phylogenetic network $\mathcal{N}'$ on X' in the following way. Take the elements of $T|C^T_{\{a,(b,c)\}}$.

      i. If there is one element in $T|C^T_{\{a,(b,c)\}}$ replace the leaf b in $\mathcal{N}'$ with this $C^T_{\{a,(b,c)\}}$-trees as a pendant tree to give $\mathcal{N}$.

      ii. If there are two elements in $T|C^T_{\{a,(b,c)\}}$ where both are caterpillar trees with one leaf, say c occurring as an element of a cherry in one and the child of the root in the other then construct a cycle by, in the second tree, joining an edge from the incoming edge of c to the incoming edge of a. Next replace the leaf b in $\mathcal{N}'$ with this uni-cyclic network on $C^T_{\{a,(b,c)\}}$ as a pendant network to give $\mathcal{N}$.

      iii. If there are two elements in $T|C^T_{\{a,(b,c)\}}$ and only at most one element is a caterpillar tree then construct a cycle by, in the one of the two trees where $\{a, b\}$ is a cherry, joining an edge from the incoming edge of c to the incoming edge of a. Next replace the leaf b in $\mathcal{N}'$ with this

uni-cyclic network on $C^T_{\{a,(b,c)\}}$ as a pendant network to give $\mathcal{N}$.

iv. Else return the statement 'T is not level-1 consistent'.

Theorem 3.4.6. *Let T be a set of phylogenetic X-trees. The algorithm* LEVEL-1 CONSTRUCT *correctly identifies in polynomial time if T is level-1 compatible. Furthermore, in the positive case* LEVEL-1 CONSTRUCT *outputs the unique standard level-1 network that displays T with minimum reticulations.*

Proof. The proof of the algorithm LEVEL-1 CONSTRUCT follows from Lemma 3.4.3 and Lemma 3.4.5. From these two lemmas it is clear that a set T of phylogenetic X-trees is level-1 compatible if and only if the following process produces a level-1 network. First, T can be level-1 reduced one pendant cluster at a time until the resulting set consists of a single element on two leaves. Second, each pendant cluster is uni-cyclic compatible and in reverse order an appropriate such uni-cyclic network is added to produce $\mathcal{N}$. It may also be noted that this process produces a standard level-1 network and hence it follows from Theorem 3.3.4 that if such a network exists $\mathcal{N}$ is the unique standard level-1 network that displays each element of T with minimum reticulations.

The running time of LEVEL-1 CONSTRUCT can be found in terms of $|X| = n$ and $|T| = k$ by observing the worst case scenario. First a pendant cluster of T is found and removed. From each leaf a in X it is checked if a occurs in a cherry in each element of T. Then for each leaf that occurs in a cherry in each element of T the cherry cluster is found and of all cherry clusters one with minimum size is selected as the pendant cluster. In worst case this gives a running time of $\mathcal{O}(nk)$. Second this process is repeated with at most n iterations. Combining these two processes together brings gives a running time of $\mathcal{O}(n^2k)$. Finally, in reverse order for each removed pendant cluster a uni-cyclic network is constructed and added to produce the output network. This process is done in constant time and hence the overall running time of the algorithm is $\mathcal{O}(n^2k)$. This running time is polynomial as required. □

3.5 Summary

In this chapter, the network-capture problem was addressed for the class of level-1 networks. For this highly structured class of networks, necessary and sufficient conditions where found for constructing a network that captures a given set of trees. Moreover, the algorithm LEVEL-1 CONSTRUCT was given that outputs the network with minimum reticulations that displays the given set of trees if such a network exists. This algorithm runs in polynomial time in the number of given trees to be displayed and the size of the leaf set.

Though confined to level-1 networks, this chapter served three important aims. First, it gave a less complex example that outlines the network-capture problem, its related attributes and potential ways to address it. Second, with this work to compare with, the following results found for more complicated classes of networks will allow the network properties that impact the network-capture problem to be specifically identified. Finally, the results found here for level-1 networks will be used and built upon to find results for tree-child networks. In following chapters uni-cyclic level-1 networks will be considered as underlying building blocks of classes of networks that are tree-based. This will allow the properties of level-1 networks to be leveraged beyond the scope of the class.

Chapter 4

Tree-Child Network Construction

4.1 Introduction

In this chapter the network-capture problem is considered in the more interesting and complicated context of tree-child networks. Tree-child networks have coincidentally been found to combine a number of important structural properties of value in phylogenetic networks. These include the already mentioned results that the class of tree-child networks is precisely the class of networks for which every vertex has a tree-path to a leaf, recall Proposition 2.0.2, every vertex is visible, recall Proposition 2.0.3, and every phylogenetic X-tree displayed is a base tree of the network, see [Sem15]. Though far less restricted than phylogenetic X-trees or even level-1 networks, the structure found in tree-child networks remains enough to, in general, give the necessary tractability to address otherwise difficult network problems. An example of this is the tree-containment problem. For an arbitrary network this problem is known to be NP-complete. However in the case of tree-child networks it was found to be polynomial time solvable, see [VISS10]. This makes tree-child networks an important class in exploring consequences of different structural properties in phylogenetic networks.

With the already well-known result that for any two given phyloge-

netic X-trees there exists a tree-child network that displays both trees (for an independent proof see Proposition 4.2.2) and that all tree-child networks are tree-based, see [FS15], it was expected that the network-capture problem would also be found to be tractable for tree-child networks. However, in this work several results demonstrate the network-capture problem for tree-child networks is significantly more complicated than for level-1 networks. The main issue comes from the fact that a tree-child network that displays two phylogenetic X-trees with a minimal number of reticulations does not necessarily display those two trees with the minimum number of reticulations, see Proposition 4.3.6. As such for tree-child networks the network-capture problem cannot be broken into a series of smaller problems considered one at a time like was found for level-1 networks. Instead it appears that every phylogenetic X-tree must be considered with respect to every other tree simultaneously. If this is true then in the context of tree-child networks, the network-capture problem would be a computationally difficult problem.

The first section of this chapter looks at the differences in how phylogenetic X-trees are displayed between level-1 networks and tree-child networks. Here two main results are identified. The first, that tree-child networks can display up to quadratically more phylogenetic X-trees than level-1 networks. This along with the fact that for any two phylogenetic X-trees there exists a tree-child network that displays both trees show that tree-child networks are significantly less restricted in their display of conflicting sets of phylogenetic X-trees, different trees on the same leaf set, than level-1 networks. However, the second result was the previously mentioned fact that a tree-child network that displays two phylogenetic X-trees with a minimal number of reticulations may not display those two trees with the minimum number of reticulations. This prevents the earlier applied strategy of breaking apart elements of the given set of trees and solving the network-capture problem piece by piece.

The second section of this chapter looks at what tree-child networks cannot display. A *universal* tree-child network on X is a tree-child network that displays every phylogenetic X-tree. Though less restricted than level-1 networks it is known that there can be no universal tree-child networks on a large number of leaves. For the set of all phy-

logenetic X-trees T_n where $|X| = n$ it was proven in [Sch70] that $|T_n| = \frac{(2n-2)!}{(n-1)!2^{n-1}}$. Whereas, it was proven in [MSW15] that for a tree-child network $\mathcal{N}$ on X that displays the maximum number of trees $|T(\mathcal{N})| \leq 2^{n-1}$. Because $\mathcal{O}(n!) >> \mathcal{O}(2^n)$ it is clear that as n becomes large there can be no tree-child network that displays all phylogenetic X-trees. By equating these two numbers it can be shown that there can be no universal tree-child network on X where $|X| \geq 4$. Hence it is known that there exist specific limits to what tree-child networks can display.

Following from the known result that for any two phylogenetic X-trees there exists a tree-child network on X that displays both trees, in this section it is proven that surprisingly the same cannot be said for any set of three phylogenetic X-trees. An example of such a set of three trees is produced and by generalising the properties that prevent this set of three trees from being simultaneously displayed by any tree-child network some of the specific limits to what tree-child networks can display are identified. Finally in Section 4.5 the results of this chapter are summarised and discussed.

4.2 Preliminaries

This section begins with independently proving two well-known results in the literature. The first shows that, unlike for level-1 networks, the restriction of a tree-child network to a subset of its leaf set may not produce a tree-child network. This difference is important to note as special care must then be taken when taking the restriction of tree-child networks. The second result is that for any two phylogenetic X-trees there exists a tree-child network on X that displays both trees. Again, this is an important difference to note between level-1 and tree-child networks. An example of two phylogenetic X-trees for which there exists no level-1 network on X that displays both is given in Fig. 4.1. This is quickly confirmed with the algorithm LEVEL-1 CONSTRUCT however in Proposition 4.2.2 it is proven that there does exist a tree-child network on X that displays both $\mathcal{T}_1$ and $\mathcal{T}_2$. Recall the solution to the network-capture problem for level-1 networks hinged on reducing

the problem to considering just two pendant trees at a time. From the outset this second result hints that the network-capture problem might be easier to address with tree-child networks however in this chapter the opposite is shown.

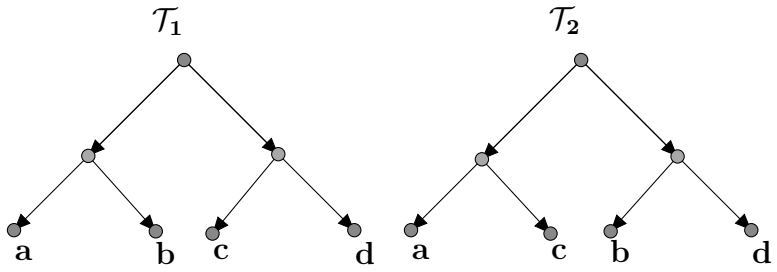

Figure 4.1: The two phylogenetic X-trees $\mathcal{T}_1$ and $\mathcal{T}_2$ are an example of two phylogenetic X-trees for which there exists no level-1 network on X that displays both.

Proposition 4.2.1. *Let $\mathcal{N}$ be a tree-child network on X with two cycles C_1 and C_2 that share v a common vertex. There exists a subset of leaves $Y \subset X$ such that the restricted network $\mathcal{N}|Y$ is not a tree-child network.*

Proof. Let $\mathcal{N}$ be a tree-child network on X with two cycles C_1 and C_2 that share v a common vertex. Let r_1 and r_2 be the reticulations that characterise C_1 and C_2 respectively. It follows from Proposition 2.0.2 that for each r_1 and r_2 there exist two non-empty mutuality exclusive sets of leaves L_1 and L_2 in $\mathcal{N}$ such that there exists a tree-path from r_1 to each element in L_1 and r_2 to each element in L_2. The vertex v is either one of r_1 or r_2 or not. In the former case, without loss of generality take r_1 to be the ancestor of r_2. For $Y = \{X - L_1\}$ the vertex r_1 in the restriction $\mathcal{N}|Y$ has no tree-path to a leaf. In the latter case, for $Y = \{L_1 \cup L_2\}$ the vertex v in the restriction $\mathcal{N}|Y$ has no tree-path to a leaf. Thus, there exists a subset of leaves $Y \subset X$ such that the restricted network $\mathcal{N}|Y$ is not a tree-child network as required.

□

Additionally the above result can be understood as the reverse of the well-understood fact that any network can be made into a tree-child network by adding leaves appropriately. The next result was mentioned without proof in [LS19].

Proposition 4.2.2. *Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be two phylogenetic X-trees. There exists a tree-child network $\mathcal{N}$ that displays both $\mathcal{T}_1$ and $\mathcal{T}_2$.*

Proof. Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be two phylogenetic X-trees. The following is a proof by induction on the number of leaves in X.

As a base case take $|X| = 1$. The trees $\mathcal{T}_1$ and $\mathcal{T}_2$ both then consist of a single vertex and furthermore may be displayed by a network $\mathcal{N}$ on X that also consists of a single vertex. As a single vertex, $\mathcal{N}$ is tree-child as required. Assume for $k \geq 1$ that if $|X| = k$ there exists a tree-child network $\mathcal{N}$ on X that displays both $\mathcal{T}_1$ and $\mathcal{T}_2$.

Now suppose that $|X| = k + 1$. Choose a leaf $a \in X$ such that a occurs in $\mathcal{T}_1$ as a part of a cherry with another leaf b. Let $Y = \{X - a\}$. The restricted trees $\mathcal{T}_1|Y$ and $\mathcal{T}_2|Y$ are both phylogenetic Y-trees where $|Y| = k$. Hence it follows from the above assumption that there exists a tree-child network $\mathcal{N}_1$ on Y that displays both $\mathcal{T}_1|Y$ and $\mathcal{T}_2|Y$.

Now attach a to $\mathcal{N}_1$ by joining it via the incoming edge on b to construct a new network $\mathcal{N}_2$ on X. As $\{a, b\}$ occurs as a cherry in $\mathcal{N}_2$ the following is true. One, every tree-path to a leaf in $\mathcal{N}_1$ is a tree-path to a leaf in $\mathcal{N}_2$. Two, every switching set that embeds $\mathcal{T}_1|Y$ in $\mathcal{N}_1$ also embeds $\mathcal{T}_1$ in $\mathcal{N}_2$. Three, if $\{a, b\}$ also occurs as a cherry in $\mathcal{T}_2$ then similarly every switching set that embeds $\mathcal{T}_2|Y$ in $\mathcal{N}_1$ also embeds $\mathcal{T}_2$ in $\mathcal{N}_2$. Thus if $\{a, b\}$ also occurs as a cherry in $\mathcal{T}_2$ then $\mathcal{N}_2$ is a tree-child network that displays both $\mathcal{T}_1$ and $\mathcal{T}_2$ as required.

On the other hand consider the case that $\{a, b\}$ does not occur as a cherry in $\mathcal{T}_2$. In $\mathcal{T}_2$ the leaf a is the sibling of a vertex v with a cluster $cl(v)$. For any embedding of $\mathcal{T}_2|Y$ in $\mathcal{N}_1$ there exists a vertex v' with a cluster $cl(v')$ such that $cl(v) = cl(v') - \{a\}$. Attach a to $\mathcal{N}_1$ as the child of a reticulation that is joined by two new edges to construct a new network $\mathcal{N}_3$ on X. The first, an edge joined to the incoming edge on b as in $\mathcal{N}_2$. The second an edge joined to the incoming edge on v', if v' is a tree vertex and v'' the child of v' otherwise. If v' is a tree vertex then every vertex in $\mathcal{N}_3$ has a tree-path to a leaf. If v' is a reticulation then v'' the child of v' is a tree vertex and $cl(v') = cl(v'')$ in $\mathcal{N}_1$ then again every vertex in $\mathcal{N}_3$ has a tree-path to a leaf.

Again every switching set that embeds $\mathcal{T}_1|Y$ in $\mathcal{N}_1$ also embeds $\mathcal{T}_1$ in

$\mathcal{N}_3$. The addition of the second new reticulation edge to the switching associated with the embedding that identified v' gives a new switching that embeds $\mathcal{T}_2$ in $\mathcal{N}_3$. Therefore by induction on $|X|$, $\mathcal{N}_3$ is a tree-child network that displays both $\mathcal{T}_1$ and $\mathcal{T}_2$ as required. □

4.3 Tree-Child Display vs Level-1 Display

To compare the differences between tree-child networks and level-1 networks this section begins by identifying how many phylogenetic X-trees can be displayed by each class of networks. Here it is conjectured that tree-child networks can display up to quadratically more trees than level-1 networks. Though a significant improvement, as already mentioned and further demonstrated by this result, this is not enough for there to be any universal tree-child networks on more then three leaves. Next in this section it is shown that unlike with level-1 networks, two phylogenetic X-trees may be displayed by a tree-child network with minimal reticulations but not necessarily the minimum possible number of reticulations.

The defining property of level-1 networks is that each cycle in a level-1 network is vertex disjoint from each other cycle. For the next two proofs the relationship of edges and paths to cycles need be qualified. Let $\mathcal{N}$ be a network on X with a cycle C. An edge uv in $\mathcal{N}$ will be called an *incoming edge* of C if v is an element of C and u is not or an *outgoing edge* of C if u is an element of C and v is not. As an extension of this, a path that finishes with an element of C and contains no other elements of C will be called and *incoming path* of C and a path that begins with an element of C and contains no other elements of C an *outgoing path* of C.

Lemma 4.3.1. *Where $|X| = n$, there exists a level-1 network that displays $2^{\frac{n-1}{2}}$ phylogenetic X-trees.*

Proof. Let $\mathcal{N}$ be a level-1 network on X that consists of a chain of 4-cycles as depicted in Fig. 4.2. Take $|X| = n$ and r to be the number of reticulations in $\mathcal{N}$. Because every cycle is vertex disjoint and there are no 3-cycles the number of distinct phylogenetic X-trees displayed by $\mathcal{N}$

is 2^r. Moreover it may be observed that $2^r = 2^{\frac{n-1}{2}}$. Hence there exists a level-1 network that displays $2^{\frac{n-1}{2}}$ phylogenetic X-trees as required. □

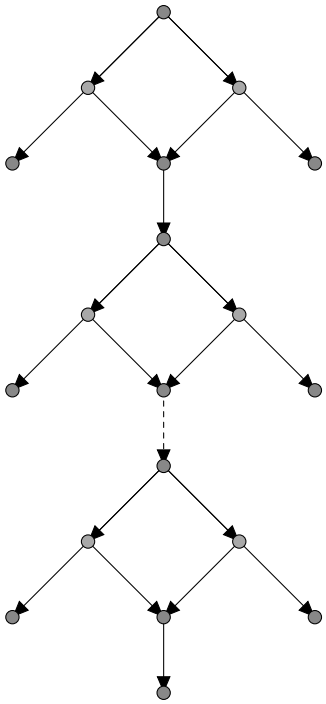

Figure 4.2: Where n is the number of leaves, the above is level-1 network that displays $2^{\frac{n-1}{2}}$ phylogenetic X-trees.

The following remains an open conjecture. Let $\mathcal{N}$ be a level-1 network on X with r reticulations and n leaves. There exists no $\mathcal{N}$ such that the number of phylogenetic X-trees displayed by $\mathcal{N}$ is greater than $2^{\frac{n-1}{2}}$.

The intuition of this conjecture comes from the understanding that the number of edges in a level-1 network is likely bounded by the size of the leaf set and the number of cycles in a level-1 network is in turn likely bounded by the number of edges. Hence to maximise the number of phylogenetic X-trees displayed by a level-1 network every edge must be either an element of a 4-cycle or joining two 4-cycles or joining a 4-cycle to a leaf.

Lemma 4.3.2. *Let $\mathcal{N}$ be a tree-child network on X with m reticulations. The network $\mathcal{N}$ displays 2^m distinct phylogenetic X-trees if and*

only if for each cycle C in $\mathcal{N}$, there exists three vertex disjoint tree-paths to a leaf that each contains precisely one element of C.

Proof. For the first direction of this proof, let $\mathcal{N}$ be a tree-child network on X with $|X| = n$ and m reticulations. The following is a proof by induction on the sum of the number of leaves $|X|$ and reticulations m in $\mathcal{N}$. As a base case let $m + n = 1$. When $m + n = 1$ the network $\mathcal{N}$ consists of a single vertex with zero reticulations and consequently zero cycles. Hence trivially for each cycle C in $\mathcal{N}$, there exists three vertex disjoint tree-paths to a leaf that each contain precisely one element of C and $\mathcal{N}$ displays $2^0 = 1$ distinct phylogenetic X-trees as required.

Where $k > 1$, assume that for $m + n = k$, if for each cycle C in $\mathcal{N}$ there exists three vertex disjoint tree-paths to a leaf that each contain precisely one element of C then $\mathcal{N}$ displays 2^m distinct phylogenetic X-trees. Now suppose that $m + n = k + 1$. It follows from Lemma 2.0.4 that there exists either a cherry or a reticulated cherry in $\mathcal{N}$.

If there exists a cherry $\{a, b\}$ in $\mathcal{N}$ then for every tree-path from a vertex v in $\mathcal{N}$ to a there also exists a tree-path from v to b. As such $\mathcal{N} \setminus a$ is a tree-child network and the sum of leaves and reticulations in $\mathcal{N} \setminus a$ is equal to k. Hence, by the above assumption if for each cycle C in $\mathcal{N} \setminus a$ there exists three vertex disjoint tree-paths to a leaf that each contain precisely one element of C then $\mathcal{N} \setminus a$ displays 2^m distinct phylogenetic X-trees. The network $\mathcal{N}$ is obtained from $\mathcal{N} \setminus a$ by joining a via an edge that joins the incoming edge on b. Thus, it can be observed that every cycle and tree-path to a leaf, other than a, starting from a cycle in $\mathcal{N}$ is a cycle and tree-path to a leaf starting from a cycle in $\mathcal{N} \setminus a$. Furthermore, the number of phylogenetic X-trees displayed by $\mathcal{N}$ is the same as the number displayed by $\mathcal{N} \setminus a$. Thus if true for $\mathcal{N} \setminus a$, then if for each cycle C in $\mathcal{N}$ there exists three vertex disjoint tree-paths to a leaf that each contain precisely one element of C then $\mathcal{N}$ displays 2^m distinct phylogenetic X-trees as required.

If there are no cherries then there exists a reticulated cherry $\{a, b\}$ in $\mathcal{N}$. Without loss of generality let b be the child of the reticulation r. Label u as the parent vertex of a and v as the other parent vertex of r. It follows from Lemma 2.0.5 that $\mathcal{N} \setminus ur$ is a tree-child network and the sum of leaves and reticulations in $\mathcal{N} \setminus ur$ is equal to k. Hence, by the

above assumption if for each cycle C in $\mathcal{N} \setminus ur$ there exists three vertex disjoint tree-paths to a leaf that each contain precisely one element of C then $\mathcal{N} \setminus ur$ displays 2^{m-1} distinct phylogenetic X-trees. The network $\mathcal{N}$ is obtained from $\mathcal{N} \setminus ur$ by adding an edge from the incoming edge on a to the incoming edge on b.

If for each cycle C in $\mathcal{N}$ containing r there exists three vertex disjoint tree-paths to a leaf that each contain precisely one element of C then a set of three such paths may be chosen that includes one path that begins with r. Let such a set of three paths terminate on x, y and $z \in X$ where x descends from r. With the addition of vr to the set, every switching in $\mathcal{N} \setminus ur$ yields the same phylogenetic X tree in $\mathcal{N}$. With the addition of ur to the set, every switching in $\mathcal{N} \setminus ur$ yields a phylogenetic X-tree where the common ancestor of x and either or both of y and z is different than every phylogenetic X-tree displayed by $\mathcal{N} \setminus ur$. Thus by induction on the sum of the number of leaves and reticulations in $\mathcal{N}$, if true for $\mathcal{N} \setminus a$, then if for each cycle C in $\mathcal{N}$ there exists three vertex disjoint tree-paths to a leaf that each contain precisely one element of C then $\mathcal{N}$ displays 2^m distinct phylogenetic X-trees as required.

For the second direction of this proof, let $\mathcal{N}$ be a tree-child network on X with m reticulations that displays 2^m distinct phylogenetic X-trees. This means each of the 2^m switchings in $\mathcal{N}$ yields a different phylogenetic X-tree. Suppose that there exists a cycle C in $\mathcal{N}$ for which there does not exist three vertex disjoint tree-paths to leaves that each contain precisely one element of C.

It follows from Lemma 3.2.1 that $\mathcal{N}$ is also a tree-child network. Hence it is known from Proposition 2.0.2 that every vertex in $\mathcal{N}$ has a tree-path to a leaf. For any cycle in $\mathcal{N}$ the peak vertex and the reticulation cannot share the same tree-path to a leaf. As such, it may be assumed that in $\mathcal{N}$ there exists two vertex disjoint tree-paths to leaves that each contain precisely one element of C. If these are the only two outgoing paths from C then C is a 3-cycle. However, it follows from Lemma 3.3.1 that if C is a 3-cycle then $\mathcal{N}$ does not display 2^m trees; a contradiction.

All outgoing edges from C not apart of the above two tree-paths must then be reticulation edges. Let r be the reticulation of C and ur and

vr be the two incoming reticulation edges on r. For any switching that does not include any of the reticulation edges outgoing from C, regardless of which of ur or vr is included the same phylogenetic X-tree is displayed; a contradiction. Therefore, if $\mathcal{N}$ displays 2^m distinct phylogenetic X-trees then there exists three vertex disjoint tree-paths to a leaf that each contain precisely one element of C as required.

□

Lemma 4.3.3. *Were $n > 1$, there exists no tree-child network that displays 2^{n-1} distinct phylogenetic X-trees.*

Proof. Let $\mathcal{N}$ be a tree-child network on leaf set X where $|X| = n$. As explained in Lemma 4.3.2 the maximum number of phylogenetic X-trees displayed by $\mathcal{N}$ is given by 2^m where m is the number of reticulation vertices in $\mathcal{N}$. Additionally, because $\mathcal{N}$ is tree-child, the root and each reticulation-vertex in $\mathcal{N}$ each has a distinct tree-path to a leaf. Hence $m \leq n - 1$ and the maximum number of phylogenetic X-trees displayed by $\mathcal{N}$ is given by 2^{n-1}. Furthermore, the n tree-paths here described are precisely the set of all maximal tree-paths in $\mathcal{N}$, each of which are vertex disjoint from the others. Suppose that $\mathcal{N}$ displays 2^{n-1} distinct phylogenetic X-trees.

Each reticulation-vertex in $\mathcal{N}$ induces a cycle. It then follows from Lemma 4.3.2 that in order for $\mathcal{N}$ to display 2^{n-1} distinct phylogenetic X-trees each cycle must have at least three outgoing tree-paths. However each tree-path in $\mathcal{N}$ is a sub-path of the n tree-paths described above. Because only one path may occur incoming to each cycle this implies that each cycle must have at least two reticulation vertices. However, for any cycle with multiple reticulations there exists one which is the descendant of all others. Consequently, no cycles staring and finishing on an ancestor reticulation contains the descendant reticulation. This means that for every reticulation there is a cycle containing at least one additional reticulation in a infinitely recurring process; a contradiction. Therefore there exists no tree-child network that displays 2^{n-1} distinct phylogenetic X-trees as required. □

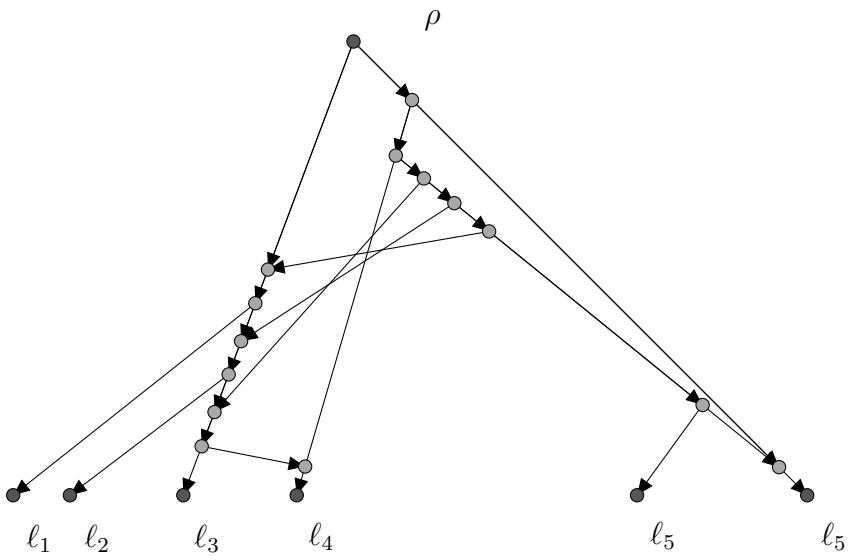

Figure 4.3: A tree-child network on $n = 6$ leaves that displays $2^{n-1} - 1$ trees.

Lemma 4.3.4. *Let $\mathcal{N}$ be a tree-child network on a leaf set X that displays a set of phylogenetic X-trees $T(\mathcal{N})$. If $|X| = n$ then $|T(\mathcal{N})| \leq 2^{n-1} - 1$ and this upper bound is sharp.*

Proof. An example of a tree-child network that displays $2^{n-1} - 1$ trees is given in Fig. 4.3 and it follows from Lemma 4.3.3 that a tree-child network can not display more. □

Proposition 4.3.5. *Let $\mathcal{N}$ be a tree-child network and $\mathcal{M}$ a level-1 network both on common leaf set X. Take $T(\mathcal{N})$ to be the set of phylogenetic X-trees displayed by $\mathcal{N}$ and $T(\mathcal{M})$ to be the set of phylogenetic X-trees displayed by $\mathcal{M}$. If both $\mathcal{N}$ and $\mathcal{M}$ display the maximum possible number of distinct phylogenetic X-trees and as conjectured the result from Lemma 4.3.1 also gives an upper bound to the number of phylogenetic X-trees that can be displayed by a level-1 network then $|T(\mathcal{N})| = |T(\mathcal{M})|^2 - 1$.*

Proof. Let $\mathcal{N}$ be a tree-child network and $\mathcal{M}$ a level-1 network both on common leaf set X where $|X| = n$. Take $T(\mathcal{N})$ and $T(\mathcal{M})$ to be the sets of phylogenetic X-trees displayed by $\mathcal{N}$ and $\mathcal{M}$ respectively. If $\mathcal{N}$ displays the maximum possible number of distinct phylogenetic X-trees

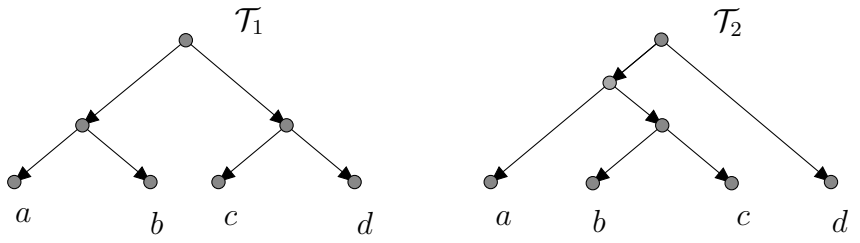

Figure 4.4: Two phylogenetic X-trees, $\mathcal{T}_1$ and $\mathcal{T}_2$.

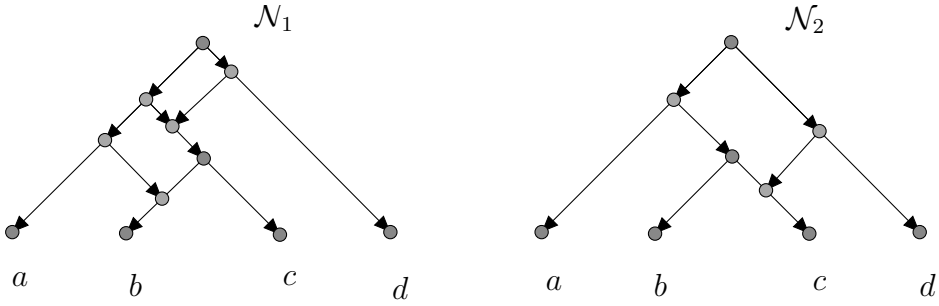

Figure 4.5: Two tree-child networks on X, $\mathcal{N}_1$ and $\mathcal{N}_2$. Both $\mathcal{N}_1$ and $\mathcal{N}_2$ display $\mathcal{T}_1$ and $\mathcal{T}_2$ from Fig. 4.4.

then it follows from Lemma 4.3.4 that $|T(\mathcal{N})| = 2^{n-1} - 1$. Similarly if $\mathcal{M}$ displays the maximum possible number of distinct phylogenetic X-trees then if as conjectured the result in Lemma 4.3.1 also gives the maximum number of phylogenetic X-trees that can be displayed by a level-1 network then $|T(\mathcal{M})| = 2^{\frac{n-1}{2}}$. Thus it is clear that $|T(\mathcal{N})| = |T(\mathcal{M})|^2 - 1$ as required. □

Proposition 4.3.6. *Let $\mathcal{N}$ be a tree-child network that displays a set T of phylogenetic X-trees with minimal reticulation vertices. The network $\mathcal{N}$ may not display T with the minimum number of reticulation vertices.*

Proof. Consider the two phylogenetic X-trees $\mathcal{T}_1$ and $\mathcal{T}_2$ in Fig. 4.4 and the two networks $\mathcal{N}_1$ and $\mathcal{N}_2$ in Fig. 4.5. First, because each vertex in $\mathcal{N}_1$ and $\mathcal{N}_2$ has a tree-path to a leaf it follows from Proposition 2.0.2 that both networks are tree-child. Second, it can be seen that both networks display $\mathcal{T}_1$ and $\mathcal{T}_2$ with a minimal number of reticulations. Third, $\mathcal{N}_2$ has one reticulation less than $\mathcal{N}_1$. Thus it is clear that

a tree-child network that displays a set of phylogenetic X-trees with a minimal number of reticulation vertices does not necessary displays the set with a minimum number of reticulation vertices as required. □

4.4 What Cannot be Displayed by a Tree-child Network

In this section the limits of what can be displayed by a tree-child network are explored. To begin with, a minor result that is expected to be already known but could not be found in the literature is independently proven. This results allows the number of vertices required to display cherries or pairs of sibling pendant trees to be counted. By a counting argument it is then proven that there exists a set of only three phylogenetic X-trees that cannot be displayed by any tree-child network. This somewhat surprising finding is believed to be new. By generalising this result Theorem 4.4.3 and Theorem 4.4.4 of this thesis are found and proven.

Embedded parent vertex. Let $\mathcal{T}$ be a phylogenetic X-tree. If for two pendant trees t_1 and t_2 in $\mathcal{T}$ the respective pendant tree roots are siblings then t_1 and t_2 *sibling pendant trees*. Additionally, for t_1 and t_2 the unique lowest common ancestor v, the parent vertex of t_1 and t_2 is quickly identifiable and unambiguous. Now let $\mathcal{N}$ be a network on X that displays $\mathcal{T}$. There may exist several vertices in $\mathcal{N}$ that can be described as a parent vertex of t_1 and t_2. To be specific, for an embedding $\mathcal{N}_\mathcal{T}$ of $\mathcal{T}$ the unique lowest common ancestor v of t_1 and t_2 in $\mathcal{N}_\mathcal{T}$ is an *embedded parent vertex* of t_1 and t_2. The vertex v is then quickly identifiable and unambiguous as the embedded parent vertex of t_1 and t_2 under the embedding $\mathcal{N}_\mathcal{T}$.

Lemma 4.4.1. *Let $\mathcal{N}$ be a tree-child network on X. Let v be an embedded parent vertex of a cherry $\{a, b\}$ occurs in a phylogenetic X-tree $\mathcal{T}$ displayed by $\mathcal{N}$. The vertex v is an embedded parent vertex of no other cherries. Moreover, every tree-path from v to a leaf in $\mathcal{N}$ finishes at one of a or b.*

Proof. Let $\mathcal{N}$ be a tree-child network on X. Let v be an embedded

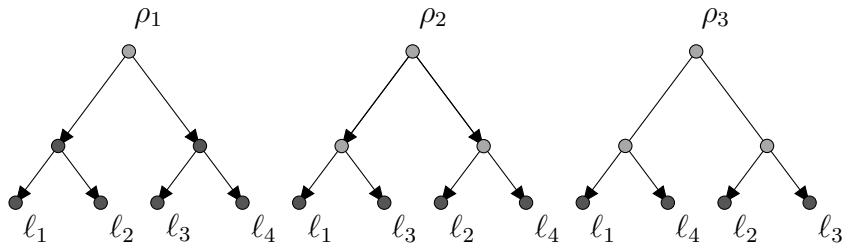

Figure 4.6: A set of three phylogenetic X-trees for which there exists no tree-child network that displays each tree.

parent vertex of a cherry $\{a, b\}$ occurs in a phylogenetic X-tree $\mathcal{T}$ displayed by $\mathcal{N}$. As all tree-paths are preserved under all embeddings and it follows from Proposition 2.0.2 that because there exists a tree-path from v to a leaf in $\mathcal{N}$ the same path exists in each embedding of $\mathcal{T}$. Hence every tree-path from v to a leaf in $\mathcal{N}$ finishes at one of a or b.

Suppose that there exists a phylogenetic X-tree $\mathcal{T}'$ such that under an embedding $\mathcal{N}_{\mathcal{T}'}$ the vertex v is an embedded parent vertex of another cherry $\{b, c\}$ in $\mathcal{T}'$. Every tree-path from v to a leaf in $\mathcal{N}$ then finishes at b. Additionally, all paths from v to a or c begin with an element of the tree-path from v to b. Because at most only one of these two sets of other paths can begin with v the vertex v cannot be the lowest common parent vertex of b and the other leaf in any embedding in $\mathcal{N}$; a contradiction.

Therefore, an embedded parent vertex v in $\mathcal{N}$ is an embedded parent vertex for precisely cherry displayed by $\mathcal{N}$. Furthermore, all tree-paths from v finish at one of a or b as required. □

Lemma 4.4.2. *There exists a set T of three phylogenetic X-trees where $|X| = 4$, such that there exists no tree-child network $\mathcal{N}$ that displays each tree in T.*

Proof. Consider the set T of three phylogenetic X-trees in Fig. 4.6. Let $\mathcal{N}$ be a tree-child network on leaf set X. Suppose that $\mathcal{N}$ displays each element of T. It follows from Lemma 4.4.1 that for each pair of cherries $\{\ell_i, \ell_j\}$ such that $\ell_i, \ell_j \in X$ there exists a distinct embedded parent vertex $v_k\{\ell_i, \ell_j\}$ in $\mathcal{N}$. This implies that in $\mathcal{N}$ there exists at least six non-leaf tree vertices of the form $v_k\{\ell_i, \ell_j\}$.

From [MSW15] it is known that the number of vertices V in a tree-child phylogenetic network is given by $V_\ell + V_r = (V+1)/2$ where V_ℓ is the number of leaves and V_r is the number of reticulation vertices. Additionally, the maximum number of reticulation vertices is given by $V_r = V_\ell - 1$. As the total number of vertices V must be a sum of V_ℓ, V_r and non-leaf tree vertices V_t the above two equations give $V_\ell - 1 = V_r = V_t - V_\ell + 1$. As $V_\ell = 4$ the the set V_t then consists precisely of the six embedded cherry parent veracities. However, this means a cherry parent vertex is the root of $\mathcal{N}$. For the root of a network to be an embedded cherry parent only the two leaves of the cherry can be displayed; a contradiction.

Therefore, there exists no rooted binary tree-child phylogenetic network $\mathcal{N}$ such that $\mathcal{N}$ displays every possible distinct pair of cherries on four leaves as required. □

Dense set of cherries. Let T be a set of phylogenetic X-trees and C a set of cherries that occur in elements of T. Take $Y \subseteq X$ to be the set of leaves that occur in elements of C. The set C will be called a *dense set of cherries* in T if for any two leaves ℓ_1 and $\ell_2 \in Y$ there exists a cherry $\{\ell_1, \ell_2\} \in C$.

Theorem 4.4.3. *Let T be a set of phylogenetic X-trees with a dense set of cherries C on a leaf subset Y where $|Y| = k$. If $k \geq 4$ then there exists no tree-child network on leaf set X such that $\mathcal{N}$ displays each element of C.*

Proof. Let T be a set of phylogenetic X-trees with a dense set of cherries C on a leaf subset Y where $|Y| = k$. Suppose that $k \geq 4$ and there exists a tree-child network $\mathcal{N}$ on leaf set $X \supseteq Y$ that displays each element of C. Let $Y = \{\ell_1, \ell_2, \ell_3, ..., \ell_k\}$. Note that C then contains at least six elements.

Because $\mathcal{N}$ is tree-child the following two points follow from Lemma 4.4.1. First there exists at least six distinct cherry parent vertices, $v_{S_1}\{\ell_1, \ell_2\}$, $v_{S_2}\{\ell_1, \ell_3\}$, $v_{S_3}\{\ell_1, \ell_4\}$, $v_{S_4}\{\ell_2, \ell_3\}$, $v_{S_5}\{\ell_2, \ell_4\}$, and $v_{S_6}\{\ell_3, \ell_3\}$ in $\mathcal{N}$. Second for each cherry parent there exists a tree-path to an element of the respective cherry. Because there are more cherry parent vertices

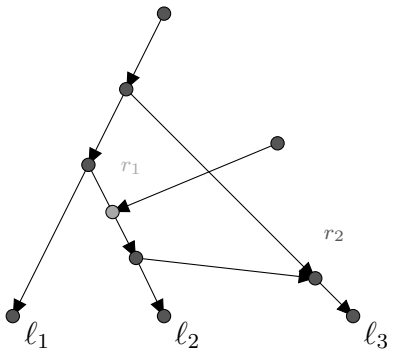

Figure 4.7: A tree-child structure that displays a dense set of three cherries.

then related leaves there must exist a tree-path to a leaf that is shared by two or more parent vertices.

Without loss of generality take ℓ_1 to occur at the termination of a tree-path common to two or more embedded parent vertices beginning with $v_{S_2}\{\ell_1, \ell_3\}$ and including $v_{S_1}\{\ell_1, \ell_2\}$. It also follows from Lemma 4.4.1 that because there can be no tree-path from $v_{S_2}\{\ell_1, \ell_3\}$ to ℓ_2 there exists a reticulation-vertex r_1 between $v_{S_1}\{\ell_1, \ell_2\}$ and ℓ_2.

Next consider the cherry $\{\ell_2, \ell_3\}$. Because the path $P(v_{S_1}\{\ell_1, \ell_3\}, \ell_1)$ is a tree-path the parent vertex $v_{S_3}\{\ell_2, \ell_3\}$ cannot occur on the path $P(v_{S_2}\{\ell_1, \ell_3\}, \ell_3)$, without preventing the cherry $\{\ell_1, \ell_3\}$ from being displayed or as an ancestor of this path without preventing the display of the cherry $\{\ell_2, \ell_3\}$. As such $v_{S_3}\{\ell_2, \ell_3\}$ is forced to descend from r_1 and reticulate with the path $P(v_{S_1}\{\ell_1, \ell_3\}, \ell_3)$ at a new reticulation-vertex r_2. This constructs a tree-child display of a dense set of cherries on three leaves. An illustration of this is given in Fig. 4.7.

Now consider the fourth leaf ℓ_4. The three cherries $\{\ell_1, \ell_4\}, \{\ell_2, \ell_4\}$ and $\{\ell_3, \ell_4\}$ must also be displayed by $\mathcal{N}$. Take first $\{\ell_3, \ell_4\}$. Again because $P(v_{S_1}\{\ell_1, \ell_3\}, \ell_1)$ is a tree-path and r_2 occurs on $P(v_{S_1}\{\ell_1, \ell_3\}, \ell_3)$ the path from parent vertex $v_{S_6}\{\ell_3, \ell_4\}$ cannot reticulate with any path from the root to ℓ_3. Hence $v_{S_6}\{\ell_3, \ell_4\}$ is forced to descend from r_2. This leaves the cherries $\{\ell_1, \ell_4\}$ and $\{\ell_2, \ell_4\}$ to be displayed. The parent vertex $v_{S_3}\{\ell_1, \ell_4\}$ must occur on the tree-path $P(v_{S_1}\{\ell_1, \ell_3\}, \ell_1)$ or above it. This means either the path $P(v_{S_3}\{\ell_1, \ell_4\}, \ell_4)$ reticulates with the

path $P(v_{S_6}\{\ell_3, \ell_4\}, \ell_4)$ or there exists a reticulation-vertex on the path $P(v_{S_6}\{\ell_3, \ell_4\}, \ell_4)$. It follows from Lemma 4.4.1 that only one of these two possibles may occur hence label this reticulation r_3.

Now consider the the cherry $\{\ell_2, \ell_4\}$. In the first above case $v_{S_5}\{\ell_2, \ell_4\}$ must descend from a reticulation-vertex, either r_1 or r_3 and there must be a reticulation edge joining $v_{S_5}\{\ell_2, \ell_4\}$ to its other cherry leaf. However no such reticulation edge may exists because $\mathcal{N}$ is tree-child; a contradiction. In the second case $v_{S_5}\{\ell_2, \ell_4\}$ must descend from r_1 or occur on the path $P(v_{S_6}\{\ell_3, \ell_4\}, \ell_4)$ and there must be a reticulation edge joining $v_{S_5}\{\ell_2, \ell_4\}$ to its other cherry leaf. However again no such reticulation edge may exists because $\mathcal{N}$ is tree-child; a contradiction. Thus no tree-child network can display these six cherries. Therefore, the largest dense set of cherries that can be displayed by a tree-child network is on three leaves as required. □

Theorem 4.4.4. *Let $\mathcal{N}$ be a tree-child network on a leaf set X such that $|X| = n$. The network $\mathcal{N}$ can display at most $2n - 3$ distinct cherries. Furthermore this upper bound is sharp as demonstrated in Fig. 4.8.*

Proof. Let $\mathcal{N}$ be a tree-child network on a leaf set X where $|X| = n$. From [MSW15] the number of leaves n, of reticulation vertices r and non-leaf tree vertices t, in $\mathcal{N}$ are related by the following formula.

$$n + r = t + 2$$

An embedded cherry parent vertex in $\mathcal{N}$ is a common ancestor of two leaves in X that becomes the lowest common ancestor under the particular embedding of a phylogenetic X-tree displayed by $\mathcal{N}$. Each embedded cherry parent vertex in $\mathcal{N}$ must be a non-leaf tree vertex in $\mathcal{N}$ as no leaf can be the ancestor of another and under an embedding each reticulation in $\mathcal{N}$ becomes a vertex with in-degree 1 and out-degree 1 and hence cannot be a lowest common ancestor of any vertex. Thus the number of cherry parent vertices p in $\mathcal{N}$ is a component of t. Rearranging the above equation for t and substituting in p gives the following inequality.

$$n + r - 2 \geq p$$

Additionally from [MSW15] the maximum number of reticulation vertices in $\mathcal{N}$ is known from

$$r_{max} = n - 1$$

Thus the maximum number of cherries possibly displayed by $\mathcal{N}$ is given by

$$2n - 3 \geq p$$

as required. □

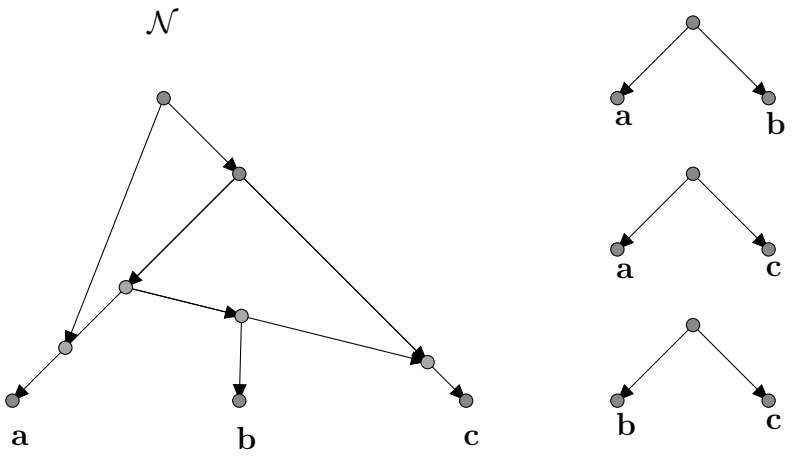

Figure 4.8: The network $\mathcal{N}$ on $X = \{a, b, c\}$ is a tree-child network that displays every possible phylogenetic X-tree where $|X| = 3$. Consequently, all three possible cherries, are displayed by $\mathcal{N}$. This gives an example that the bound in Theorem 4.4.4 is sharp.

4.5 Summary

In this chapter the network-capture problem was considered for the class of tree-child networks. Unlike for level-1 networks, it was identified that the network-capture problem could not be directly boot-strapped from constructing networks that display pairs of pendant trees from the given set of phylogenetic X-trees. This was surprising as like level-1 networks, tree-child networks are tree-based and it is well-known that for any two phylogenetic X-trees there exists a tree-child network on X that displays both trees. Critically it was found from Proposition 4.3.6 that unlike for level-1 networks a tree-child network that displays a set of phylogenetic X-trees with minimal reticulations may not display the set with minimum reticulations. This means that for the given set of phylogenetic X-trees, in the network-capture problem, each tree must be considered with respect to every other tree. Moreover, if a network is found that can display each tree in the given set it may not display the set with the minimum number of reticulations.

The subsequent work in this chapter looked at other limits to what can be displayed by a tree-child network. Again surprisingly, though for any two phylogenetic X-trees there exists a tree-child network that displays both trees it was found that the same cannot be said for a set of three phylogenetic X-trees. Here a set of three trees where found that cannot be displayed by any tree-child network. This was then generalised to combinations of cherries that can not be displayed by tree-child networks.

Though tree-child networks are much less restricted than level-1 networks in what they can display it is well known that significant limits exists as there are no tree-child universal networks where $|X| > 3$. An example of a universal tree-child network on $|X| = 3$ is given in Fig. 4.8.

To further address the network-capture problem for tree-child networks future work might look into quantifying how many ways two trees can be displayed by a tree-child network with minimal reticulations. If this number was readily knowable and happens to grow slowly perhaps a bootstrap method might yet be found for the network-capture problem.

Chapter 5

Normal Network Reconstruction

5.1 Introduction

In this chapter the network-capture problem is readdressed for the class of tree-child networks. This time, two assumptions are made to give greater tractability to the problem. The first assumption is that for the particular given set T of phylogenetic X-trees, there exist a tree-child network $\mathcal{N}$ on X that displays T. The second is that there are no short-cut edges in $\mathcal{N}$. The class of tree-child networks that contain no short-cut edges was introduced in [Wil07]. Originally this class of networks was referred to as 'regular' networks however to avoid confusion with the existing class of the same name defined in [BSS05], the class of tree-child networks that contain no short-cut edges is now known as *normal networks*. By these assumptions the network-capture problem is then reduced to that of reconstructing a normal network $\mathcal{N}$ on X from a set $T \subseteq T(\mathcal{N})$.

The same variation of the network-capture problem was considered in [Wil11]. In this paper Willson demonstrated that for a normal network $\mathcal{N}$ on X and a special set of phylogenetic X-trees $T \subseteq T(\mathcal{N})$, that $\mathcal{N}$ is the unique network on X that displays each element of T with minimal reticulations. Moreover, Willson proved that there exists a polynomial time algorithm that with input T would output $\mathcal{N}$.

In this chapter independent proofs are given for both of the results mentioned above. Moreover, the method developed here improves on the work published in [Wil11] by reducing the required size of the set T and the operational time of the algorithm required to construct $\mathcal{N}$ from T.

For a normal network $\mathcal{N}$ on X where $|X| = n$ it is proven that there exists a set $T \subseteq T(\mathcal{N})$ where $|T| = n+1$ for which $\mathcal{N}$ is the unique normal network that displays T with minimal reticulations and a polynomial time algorithm exists that reconstructs $\mathcal{N}$ from T. Willson in comparison requires for the same purpose, a set $T' \subseteq T(\mathcal{N})$ where $|T'| = \mathcal{O}(n^2)$. This is because the method in [Wil11] depends on the number of edges in the network where the method produced here depends on the number of reticulation vertices. Though Willson approximates the number of edges in a normal network to grow quadratically this can be further refined. In [MSW15], where r is the number of reticulation vertices and n is the number of leaves, the number of edges in a network is found to be given by $3r + 2n - 2$. Again from [MSW15], the number of reticulation vertices in a normal network is known to be bounded by $r \leq n-2$. Together this means that the number of phylogenetic X-trees required to reconstruct a normal network using Willson's method is at most $5n-8$. When n is large this is approximately five times larger then the method below which using the same relationships found in [MSW15] requires at most a set of $n+1$ phylogenetic X-trees.

This chapter is organised as follows. First in Section 5.2 the formal definitions and known results from the literature are given that are required to prove the main results of the chapter. Then in Section 5.3 the main results of this chapter, Theorem 5.3.5 and Corollary 5.3.6 are proven. Finally, in Section 5.4 the results of this chapter are summarised and discussed.

5.2 Preliminaries

This section will begin by independently proving several initial results to provide a starting point for this work. Particularly, the important known result that no phylogenetic X-tree can be embedded in a nor-

mal network in more than one way will be independently proven. It is specifically because of this result that the assumption that there are no short-cut edges in the tree-child network is made. Knowing that no phylogenetic X-tree is displayed more than once greatly reduces the potential complexity of the problem and provides the required tractability.

The first of the following results quickly identifies that for a tree-child networks all switchings may be assumed to embed a phylogenetic X-tree. The second demonstrates that if a tree-child network displays less than the maximum number of phylogenetic X-trees for the number of its reticulations then it is not a normal network. Finally, the third specifies a particular deletion operation that when applied to normal networks produces another normal network.

Lemma 5.2.1. *Let $\mathcal{N}$ be a tree-child network on X. Every switching in $\mathcal{N}$ embeds a phylogenetic X-tree.*

Proof. Let $\mathcal{N}$ be a tree-child network on X with root ρ. Suppose there exist a switching S in $\mathcal{N}$ that does not embed a phylogenetic X-tree.

Delete each reticulation edge in $\mathcal{N}$ that is not in S and label the resulting network $\mathcal{N}_S$. For each reticulation-vertex there is now precisely one incoming edge. The resulting directed graph then contains no cycles. Furthermore, because each parent vertex of a reticulation in $\mathcal{N}$ is visible every directed path from ρ terminates at a leaf $\ell \in X$. Suppressing all internal degree-2 vertices then yields a phylogenetic X-tree $\mathcal{T}$. Hence S embeds a phylogenetic X-tree; a contradiction. Therefore, every switching in a tree-child network embeds a phylogenetic X-tree as required. □

Lemma 5.2.2. *Let $\mathcal{N}$ be a tree-child network on X with r reticulations and tree display set $T(\mathcal{N})$. If the number of trees $|T(\mathcal{N})| < 2^r$ then there exists at least one short-cut edge in $\mathcal{N}$.*

Proof. Let $\mathcal{N}$ be a tree-child network on X with r reticulations and tree display set $T(\mathcal{N})$. Suppose that $|T(\mathcal{N})| < 2^r$ and there exists no short-cut edges in $\mathcal{N}$.

First note the following is known. Every embedded tree in a tree-child network contains every vertex of the network. Every vertex has a tree-

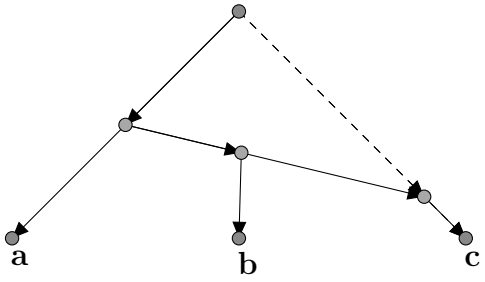

Figure 5.1: A tree-child network that displays 2^r where $r = 1$ distinct phylogenetic X-trees and has a short-cut edge marked by the dashed lines.

path to a leaf. Every tree-path in a tree-child network is a path in every embedded tree in the network.

As $|T(\mathcal{N})| < 2^r$ there exits at least two embedded trees, $\mathcal{N}_{\mathcal{T}_1}$ and $\mathcal{N}_{\mathcal{T}_2}$ in $\mathcal{N}$ that both display the same phylogenetic X-tree, $\mathcal{T}$. As $\mathcal{N}_{\mathcal{T}_1} \neq \mathcal{N}_{\mathcal{T}_2}$ there exists a reticulation $x \in \mathcal{N}$ with parent vertices u and v and child vertex w such that without loss of generality ux is an element of $\mathcal{N}_{\mathcal{T}_1}$ and vx is an element of $\mathcal{N}_{\mathcal{T}_2}$. Because $\mathcal{N}_{\mathcal{T}_1}$ and $\mathcal{N}_{\mathcal{T}_2}$ both display $\mathcal{T}$, ignoring all degree two vertices the set of paths from the root to each leaf in X are isomorphic up to relabelling for both $\mathcal{N}_{\mathcal{T}_1}$ and $\mathcal{N}_{\mathcal{T}_2}$.

Hence regardless of the choice of ux or vx the pendant tree descending from u in $\mathcal{N}_{\mathcal{T}_1}$ is isomorphic up to relabelling to the pendant tree descending from v in $\mathcal{N}_{\mathcal{T}_2}$. This includes at least three tree-paths to leaves P_u, P_v and P_w, starting from u, v and w respectively. This means one of P_u and P_v is a sub-path of the other in $\mathcal{N}$ and ignoring degree-2 vertices in $\mathcal{N}_{\mathcal{T}_1}$ and $\mathcal{N}_{\mathcal{T}_2}$ both paths are isomorphic up to relabelling. Thus u, v and w together comprise a three-cycle with a short-cut edge; a contradiction. Therefore, if $|\mathcal{T}| < 2^r$ then there exists a short-cut edge in $\mathcal{N}$ as required. □

Remark. There exists a tree-child network that displays 2^r distinct phylogenetic X-trees and has a short-cut edge. An example of such a network is given in Fig. 5.1.

Lemma 5.2.3. *Let $\mathcal{N}$ be a normal network on X with a reticulation edge ur. The network $\mathcal{N} \setminus ur$ is also a normal network.*

Proof. Let $\mathcal{N}$ be a normal network on X. Suppose that there exists a reticulation edge ur in $\mathcal{N}$ such that $\mathcal{N} \setminus ur$ is not a normal network.

Because all normal networks are tree-child networks it follows from Lemma 2.0.5 that $\mathcal{N} \setminus ur$ is tree-child. In order for $\mathcal{N} \setminus ur$ to a tree-child network but not a normal network it must have a short-cut xy. For such an edge to occur there exists two paths $P(x, y)$ and $P_u(x, y)$ from x to y in $\mathcal{N}$ and $P_u(x, y) = x, u, y$. Hence in $\mathcal{N} \setminus ur$ there would exist a short-cut edge xy. However, the vertex u in $\mathcal{N}$ has two reticulations r and y as children and as such is not a normal network; a contradiction. Therefore, if a network $\mathcal{N}$ is a normal network and ur is a reticulation edge in $\mathcal{N}$ then the network $\mathcal{N} \setminus ur$ is also a normal network as required. □

Finally, here an independent proof is given for the known result that normal networks do not display any phylogenetic X-tree more than once. This result was first proven as Corollary 2 in [CLS14].

Lemma 5.2.4. *Let $\mathcal{N}$ be a normal network on X. There exists no two embedded trees in $\mathcal{N}$ that display the same phylogenetic X-tree.*

Proof. Let $\mathcal{N}$ be a normal network with r reticulations. Suppose that there exists two embedded trees in $\mathcal{N}$ that display the same phylogenetic X-tree. Hence $|T(\mathcal{N})| < 2^r$. It follows from Lemma 5.2.2 that $\mathcal{N}$ has a short-cut edge; a contradiction. Therefore, there exists no $\mathcal{T}_i$, $\mathcal{T}_j \in T(\mathcal{N})$ such that $\mathcal{T}_i = \mathcal{T}_j$ as required. □

5.3 Normal Network Reconstruction

In this section it will be demonstrated that a normal network may be reconstructed from a linear subset of the phylogenetic X-trees it displays. To achieve this, results already obtained for level-1 networks and tree-based networks will be extended on and employed.

Complement tree. A normal network is by definition also a tree-child network and as identified in [FS15] then also a tree-based network. It follows from Lemma 5.2.1 that, for any phylogenetic X-tree embedded in a normal network, the complement switching embeds a phylogenetic X-tree also displayed by the network. Let $\mathcal{N}$ be a network on X. Two phylogenetic X-trees $\mathcal{T}_1$ and $\mathcal{T}_2$ that are displayed by $\mathcal{N}$ via a switching and its complement will be called a *complement pair* and each tree a *complement tree* with respect to the other.

Cover-set. Next, let $\mathcal{N}$ be a tree-child network. Take $\mathcal{T}_0$ to be a phylogenetic X-tree that is displayed by $\mathcal{N}$ via a switching set S. It follows from Lemma 5.2.1 that every switching in $\mathcal{N}$ yields a phylogenetic X-tree displayed by $\mathcal{N}$. A *cover-set* F_0 of $\mathcal{T}_0$ in $\mathcal{N}$ is the set of phylogenetic X-trees that are displayed by $\mathcal{N}$ each via a switching that differs from S by precisely one element. Together a base tree $\mathcal{T}_0$ and corresponding cover-set F_0 in a network and will be referred to as a *base and its cover-set*.

Importantly, for a tree $\mathcal{T}_0$ and an element of its cover-set $\mathcal{T}_1$ in a tree based network $\mathcal{N}$ it may be observed that $\mathcal{T}_0$ and $\mathcal{T}_1$ are uni-cyclic compatible. Moreover, there exists a set of embeddings of $\mathcal{T}_0$ and each element of its cover-set that collectively contain every reticulation in $\mathcal{N}$. because of this a base and cover-set of a tree-child network is related by a series of level-1 networks. This means the properties already found that determine how and when phylogenetic X-trees may be displayed by level-1 networks may be utilised.

Lemma 5.3.1. *Let $\mathcal{N}$ be a normal network on X. For each phylogenetic X-tree $\mathcal{T}_0 \in T(\mathcal{N})$ there exists a unique complement base tree $\bar{\mathcal{T}}_0 \in T(\mathcal{N})$ and cover-set $F_0 \subseteq T(\mathcal{N})$.*

Proof. Let $\mathcal{N}$ be a normal network on X that displays a phylogenetic X-tree $\mathcal{T}_0$. As all normal networks are tree-child it follows from Lemma 5.2.1 that every switching in $\mathcal{N}$ yields a phylogenetic X-tree displayed by $\mathcal{N}$. Additionally, it follows from Lemma 5.2.4 that no two switchings in $\mathcal{N}$ yield the same phylogenetic X-tree. Let S be the switching that yields $\mathcal{T}_0$. The complement switching $\bar{S}$ and every switching that differs from S by precisely one element then each yield

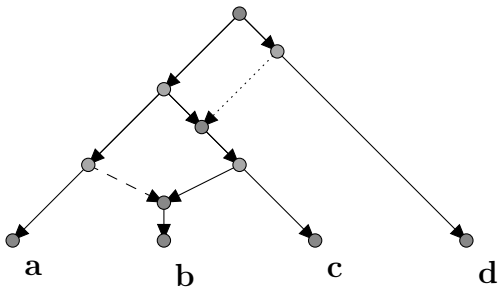

Figure 5.2: Base and Cover Set.
Here a normal network is presented with a base tree shown in solid lines. By separately adding the dotted or dashed reticulation edge to the base tree two level-1 networks that each display precisely two trees may be observed. These trees that are displayed by alternating precisely one reticulation edge of the base tree are the cover trees that correspond this base tree in this network. Note for any tree-based network for a given base tree there exists r corresponding embedded cover trees where r is the number of reticulation vertices in the network.

a phylogenetic X-tree displayed by $\mathcal{N}$. Call the former phylogenetic X-tree $\bar{\mathcal{T}}_0$ and the later set of phylogenetic X-trees F_0.

Because $\bar{S}$ is the unique complement switching for S in $\mathcal{N}$ and $\bar{S}$ is the only switching that yields $\bar{\mathcal{T}}_0$ the phylogenetic X-tree $\bar{\mathcal{T}}_0$ is the unique complement tree of $\mathcal{T}_0$ displayed by $\mathcal{N}$. Analogously, the set of phylogenetic X-trees in F_0 is the unique set of phylogenetic X-trees yielded by switchings that differ from S by precisely one element.

Finally, because all normal networks are tree-child it follows from [Sem15] that both $\mathcal{T}_0$ and $\bar{\mathcal{T}}_0$ are base trees of $\mathcal{N}$. Therefore, For each phylogenetic X-tree $\mathcal{T}_0 \in T(\mathcal{N})$ there exists a unique complement base tree $\bar{\mathcal{T}}_0 \in T(\mathcal{N})$ and cover-set $F_0 \subseteq T(\mathcal{N})$ as required. □

Lemma 5.3.2. *Let $\mathcal{N}$ be a normal network on X with at least two reticulations that displays a phylogenetic X-tree $\mathcal{T}_0$ with complement tree $\bar{\mathcal{T}}_0$ and cover-set F_0. There exists no two phylogenetic X-trees $\mathcal{T}_1, \mathcal{T}_2 \in \mathcal{T}_0 \cup \bar{\mathcal{T}}_0 \cup F_0$ such that $\mathcal{T}_1 = \mathcal{T}_2$.*

Proof. Let $\mathcal{N}$ be a normal network on X that displays a phylogenetic X-tree $\mathcal{T}_0$ with complement tree $\bar{\mathcal{T}}_0$ and cover-set F_0. Suppose that there exists two phylogenetic X-trees $\mathcal{T}_1, \mathcal{T}_2 \in \mathcal{T}_0 \cup \bar{\mathcal{T}}_0 \cup F_0$ such that $\mathcal{T}_1 =$

$\mathcal{T}_2$. It follows from [CLS14] that no phylogenetic X-tree is displayed more than once by $\mathcal{N}$; a contradiction. Therefore, there exists no two phylogenetic X-trees $\mathcal{T}_1, \mathcal{T}_2 \in \mathcal{T}_0 \cup \bar{\mathcal{T}}_0 \cup F_0$ such that $\mathcal{T}_1 = \mathcal{T}_2$ as required. □

By looking at just a base and cover-set of a normal network it may be identified if a network has a cherry or a reticulated cherry. This is because either every element of the set must have a common cherry or all but one shares a common cherry with the base-tree.

Lemma 5.3.3. *Let $\mathcal{N}$ be a normal network on X with a reticulated cherry $\{a, b\}$. Label the peak vertex of the reticulated cherry u, the reticulation w, the other parent vertex of the reticulation v and without loss of generality let b be the child of the reticulation. A tree displayed by $\mathcal{N}$ has the cherry $\{a, b\}$ if and only if the embedded tree contains the edge uw.*

Proof. Let $\mathcal{N}$ be a normal network on X with a reticulated cherry $\{a, b\}$. Label the peak vertex of the reticulated cherry u, the reticulation w, the other parent vertex of the reticulation v and without loss of generality let b be the child of the reticulation.

Suppose there exists an embedding of a phylogenetic X-tree $\mathcal{T}$ with the cherry $\{a, b\}$ that does not contain the edge uw. There then exists another embedded parent vertex u' of a and b. Because all normal networks are tree-child it follows from Proposition 2.0.2 that there exists at least one tree-path to a leaf for both u and u'. Moreover, because there are no reticulation edges in a tree-path, every embedded tree in $\mathcal{N}$ contains every tree-path in $\mathcal{N}$. Hence, there exists precisely one tree-path from u' to a leaf in $\mathcal{N}$, that leaf must be a and that tree-path must also contain u.

Because the path from u' to b does not contain uw an alternate path from u' to w exists. However, because $\mathcal{N}$ has no short-cut edges this alternate path from u' to w must contain another vertex x. It then follows again from Proposition 2.0.2 that there exists a tree-path from x to a leaf $c \neq a, b$. Again this tree-path from x to c occurs in every embedded tree in $\mathcal{N}$. Thus u' is not a embedded parent of a and b; a contradiction.

Suppose conversely that there exists an embedding of a phylogenetic X-tree $\mathcal{T}$ that contains the edge uw but does not have the cherry $\{a, b\}$. As there exists a tree-path from u to a and from w to b and all tree-paths in $\mathcal{N}$ occur in every embedded tree in $\mathcal{N}$ the phylogenetic X-tree $\mathcal{T}$ has the cherry $\{a, b\}$; a contradiction. Therefore, a tree displayed by $\mathcal{N}$ has the cherry $\{a, b\}$ if and only if each embedding of the tree contains the edge uw as required. □

Lemma 5.3.4. *Let $\mathcal{N}$ be a normal network on X and $\mathcal{T}_0$ a base tree of $\mathcal{N}$ with cover-set F_0. Let $a, b \in X$ be the leaves of either a cherry or a reticulated cherry in $\mathcal{N}$. If $\mathcal{T}_0$ has the cherry $\{a, b\}$, either every or all but precisely one of the phylogenetic X-trees in F_0 has the cherry $\{a, b\}$. If $\mathcal{T}_0$ does not have the cherry $\{a, b\}$ then precisely one of the phylogenetic X-trees in F_0 has the cherry $\{a, b\}$.*

Proof. Let $\mathcal{N}$ be a normal network on X and $\mathcal{T}_0$ a base tree of $\mathcal{N}$ with cover-set F_0. Because all normal networks are tree-child it follows from Lemma 2.0.4 that there exists two leaves a and b in X that occur as the leaves of either a cherry or a reticulated cherry in $\mathcal{N}$. If $\{a, b\}$ is a cherry in $\mathcal{N}$ then $\{a, b\}$ is a cherry in every phylogenetic X-tree displayed by $\mathcal{N}$. Hence, in $\mathcal{T}_0$ and every element of F_0 the leaves a and b occur as a cherry.

Alternately, consider the case that a and b occur as the leaves of a reticulated cherry in $\mathcal{N}$. As such, there exists an embedded parent vertex v and a reticulation-vertex r on the shortest up-down path from a to b in $\mathcal{N}$. It follows from Lemma 5.3.3 that a tree displayed by $\mathcal{N}$ has the cherry $\{a, b\}$ if and only if each embedding of the tree contains the edge vr. Let S be the switching that yields $\mathcal{T}_0$. The set of phylogenetic X-trees in the cover-set F_0 is the set of phylogenetic X-trees displayed by $\mathcal{N}$ that are yielded by a switching that differs from S by precisely one element. Hence, if $\{a, b\}$ is a cherry in $\mathcal{T}_0$ there exists precisely one phylogenetic X-tree in F_0 whose switching does not contain the edge vr and consequently does not contain the cherry $\{a, b\}$.

Alternatively, if $\{a, b\}$ is not a cherry in $\mathcal{T}_0$ there exists precisely one phylogenetic X-tree in F_0 that's switching contains the edge vr and consequently contains the cherry $\{a, b\}$. Therefore, if $\mathcal{T}_0$ has the cherry

$\{a, b\}$, either every or all but precisely one of the phylogenetic X-tree in F_0 has the cherry $\{a, b\}$ and if $\mathcal{T}_0$ does not have the cherry $\{a, b\}$ then precisely one of the phylogenetic X-tree in F_0 has the cherry $\{a, b\}$ as required. □

Together, these results facilitate the proof of the main result of this chapter, Theorem 5.3.5. That is, for a normal network $\mathcal{N}$ there exists a set of phylogenetic X-trees T displayed by $\mathcal{N}$ such that $|T| \leq |X|+1$ and $\mathcal{N}$ is the unique normal network that displays T with minimal reticulations.

Theorem 5.3.5. *Let $\mathcal{N}$ be a normal network on X with r reticulations. Take $\mathcal{T}_0$ to be a base tree with complement $\bar{\mathcal{T}}_0$ and cover set F_0 in $\mathcal{N}$. Where $T = \{\mathcal{T}_0, \bar{\mathcal{T}}_0\} \cup F_0$ and $|T| = r + 2$ the network $\mathcal{N}$ is the unique normal network on X that displays each element of T with minimal reticulations up to isomorphism.*

Proof. Let $\mathcal{N}$ be a normal network on X with r reticulations. Take any phylogenetic X-tree $\mathcal{T}_0$ displayed by $\mathcal{N}$. Because all normal networks are tree-child it follows from [Sem15] that $\mathcal{T}_0$ is a base tree of $\mathcal{N}$ and from Lemma 5.3.1 there exists a unique complement tree $\bar{\mathcal{T}}_0$ and cover-set F_0 all displayed by $\mathcal{N}$. Let $T = \{\mathcal{T}_0 \cup \bar{\mathcal{T}}_0 \cup F_0\}$. Because $|F_0| = r$ it may be immediately observed that $|T| = r + 2$.

The following is an inductive proof on the sum of reticulations and leaves in $\mathcal{N}$. When $r + |X| = 1$, the network $\mathcal{N}$ is a single vertex. As such $\bar{\mathcal{T}}_0 \cup F_0 = \emptyset$ and $\mathcal{N}$ uniquely displays each element of T, the single vertex $\mathcal{T}_0$. Assume that for $r + |X| = k$ the network $\mathcal{N}$ is the unique normal network that displays each element of T. Now for $r+|X| = k+1$ suppose that there exists another normal network $\mathcal{M}$ on X that also displays each element of T with minimal reticulations. It follows again from [Sem15] that $\mathcal{T}_0$ is also a base tree of $\mathcal{M}$.

From [Sem15] and Lemma 2.0.4 it follows that there exists two leaves a and b in X that occur as the leaves of either a cherry or a reticulated cherry in $\mathcal{M}$. Consider the case that a and b occur as a cherry in $\mathcal{M}$. Every phylogenetic X-tree displayed by $\mathcal{M}$, specifically every element of T, then also has the cherry $\{a, b\}$. Because $\mathcal{M}$ is a normal network for every vertex in $\mathcal{M}$ there exists a tree-path from that vertex to a

leave and there exists no short-cut edges in $\mathcal{M}$. As $\{a, b\}$ occurs as a cherry in $\mathcal{M}$ for every tree-path from a vertex to a there exists a tree-path from that vertex to b. Hence, it may be immediately observed that for every vertex in $\mathcal{M} \setminus a$ there exists a tree-path to a leaf and because every edge in $\mathcal{M} \setminus a$ is also an edge in $\mathcal{M}$ there are no short-cut edges in $\mathcal{M} \setminus a$. Thus, $\mathcal{M} \setminus a$ is a normal network and it is clear that $\mathcal{M} \setminus a$ displays each element of $T \setminus a$. In the same way $\mathcal{N} \setminus a$ is also a normal network that displays each element of $T \setminus a$. As $r + |X| - 1 = k$ by the above assumption $\mathcal{N} \setminus a$ is the unique normal network that displays $T \setminus a$. Hence $\mathcal{M} \setminus a = \mathcal{N} \setminus a$. Next, with a minimal number of reticulation edges, join a to $\mathcal{N} \setminus a$ to display the elements of T. Because $\{a, b\}$ is a cherry in each phylogenetic X-tree in T this can only be done by adding a single edge joining a to the incoming edge of b. Thus, $\mathcal{N} = \mathcal{M}$; a contradiction.

Next consider the case that a and b occur as a reticulated cherry in $\mathcal{M}$. Label the peak vertex u, the parent vertex of u as u', the reticulation w, the other parent of w as v, the parent vertex of v as v' and without loss of generality let w be the parent of b.

It follows from Lemma 5.3.4 and Lemma 5.3.3 that if $\{a, b\}$ is a cherry in $\mathcal{T}_0$ then $\{a, b\}$ is a cherry in all but one of the elements of F_0 and if $\{a, b\}$ is not a cherry in $\mathcal{T}_0$ then $\{a, b\}$ is a cherry in only one of the elements of F_0. If $\{a, b\}$ is a cherry in $\mathcal{T}_0$ take $\mathcal{M} \setminus vw$ otherwise take $\mathcal{M} \setminus uw$. It follows from Lemma 5.2.3 that both $\mathcal{M} \setminus vw$ and $\mathcal{M} \setminus uw$ are normal networks. It follows from Lemma 5.3.3 that every embedded tree in $\mathcal{M}$ that has the cherry $\{a, b\}$ contains the edge uw and no embedded tree in $\mathcal{M}$ that has the cherry $\{a, b\}$ contains the edge vw. Hence, the network selected displays all but one element of F_0.

Label the phylogenetic X-tree in F_0 not displayed by the selected network as $\mathcal{T}_1$ and the switching of $\mathcal{T}_1$ in $\mathcal{N}$ as S_1. Where S is the switching of $\mathcal{T}_0$ in $\mathcal{N}$, S_1 has precisely one element e such that $e \notin S$. As $r + |X| - 1 = k$ by the above assumption $\mathcal{N} \setminus e$ is the unique normal network that displays $T \setminus \mathcal{T}_1$. Hence, the appropriate choice of $\mathcal{M} \setminus vw$ or $\mathcal{M} \setminus uw$ is equal to $\mathcal{N} \setminus e$.

By joining a single edge to $\mathcal{N} \setminus e$ the network $\mathcal{M}$ can be re-obtained.

Consider $\mathcal{T}_0$ and $\mathcal{T}_1$. As $\mathcal{T}_1 \in F_0$ there exists a normal uni-cyclic network $\mathcal{U}$ that displays precisely $\mathcal{T}_0$ and $\mathcal{T}_1$. Moreover, because there can be no short-cut edges in a normal network it follows from Theorem 3.3.4 that $\mathcal{U}$ is the unique standard level-1 network that displays $\mathcal{T}_0$ and $\mathcal{T}_1$ with the minimum number of reticulations. Hence, where a' is the parent vertex of a and b' is the parent vertex of b in $\mathcal{T}_0$ there exists precisely one reticulation edge that joins two the edges $a'a$ and $b'b$ in $\mathcal{T}_0$ in precisely one way to produce $\mathcal{U}$. Because $\mathcal{T}_0$ a base tree of $\mathcal{N}, \mathcal{M}$ and $\mathcal{U}$ it then follows that there exists only one way a single edge can be added to $\mathcal{N} \setminus e$ to produce a normal network on X that displays T with minimal reticulations. Thus up to isomorphism $\mathcal{N} = \mathcal{M}$; a contradiction. Therefore, by induction on the size of $r + |X|$, there exists a set of phylogenetic X-trees $T \subseteq T(\mathcal{N})$ where $|T| = r + 2$, such that $\mathcal{N}$ is the unique normal network on X that displays each element of T with minimal reticulations as required.

□

Finally, it is proven in the next result that there exists a polynomial time algorithm that reconstructs a normal network $\mathcal{N}$ from a subset of $T(\mathcal{N})$ that grows linearly with respect to $|X|$. This is done by observing that the induction process employed in the proof of Theorem 5.3.5 may be adapted into an algorithm.

Corollary 5.3.6. *Let $\mathcal{N}$ be a normal network on X and $\mathcal{T}_0$ a phylogenetic X-tree displayed by $\mathcal{N}$. For $\mathcal{T}_0$ there exists a complement tree $\bar{\mathcal{T}}_0$ and cover set F_0. Let $T = \mathcal{T}_0 \cup \bar{\mathcal{T}}_0 \cup F_0$. There exists a polynomial time algorithm that reconstructs $\mathcal{N}$ from T.*

Proof. Let $\mathcal{N}$ be a normal network on X and $\mathcal{T}_0$ a phylogenetic X-tree displayed by $\mathcal{N}$. Because all normal networks are tree-child it follows from [Sem15] that $\mathcal{T}_0$ is a base tree of $\mathcal{N}$ and from Lemma 5.3.1 that for $\mathcal{T}_0$ there exists a unique complement tree $\bar{\mathcal{T}}_0$ and cover set F_0 all displayed by $\mathcal{N}$. Let $T = \mathcal{T}_0 \cup \bar{\mathcal{T}}_0 \cup F_0$. It follows from Theorem 5.3.5 that $\mathcal{N}$ is the unique normal network that displays T with minimal reticulations. Additionally from the inductive proof of Theorem 5.3.5 an algorithm is given that constructs $\mathcal{N}$ from T that operates on the

number of leaves in X plus the reticulations in $\mathcal{N}$. Because $\mathcal{N}$ is a normal network it follows from [MSW15] that if $|X| = n$ then the number of reticulations in $\mathcal{N}$ has a sharp upper bound of $n-2$. Thus the maximum running time of this algorithm is $\mathcal{O}(X)$. Therefore, there exists a polynomial time algorithm that reconstructs $\mathcal{N}$ from T as required. □

5.4 Summary

In this chapter the network-capture problem was readdressed for the class of tree-child networks. With the benefit of the following two assumptions the problem was made tractable. The first, that for a given set of phylogenetic X-trees there exists a tree-child network that displays the set. The second, that the tree-child network has no short-cut edges and hence is a normal network. With these assumptions the problem became, can a normal network be faithfully reconstructed from a subset of phylogenetic X-trees it displays. Here it was found that a normal network can be faithfully reconstructed from a subset of the phylogenetic X-trees it displays that grows linearly with respect to the number of leaves in the network. This improved on the results published in [Wil11] where Willson required a subset of the phylogenetic X-trees displayed that grows quadratically with respect to the number of leaves in the network.

The main results proven in this chapter used the tree-based property of tree-child networks and identified specific uni-cyclic networks that act as underlying building blocks for a given normal network. As uni-cyclic networks are level-1 networks, the properties found in Chapter 3 could be directly applied. Specifically Theorem 3.3.4, that the set of trees displayed by a standard level-1 network with minimal reticulations is uniquely displayed by that standard level-1 network. With this, a base tree, cover-set and complement base tree of a given normal network can be quickly found that is uniquely displayed by the network and can be used to reconstruct the network in linear-time.

Future work could look at adapting and extending this approach to tree-child networks. This would require untangling the complications that are caused by short-cut edges and consequently phylogenetic X-trees

that potentially are displayed multiple times. If successful for tree-child networks, then this could then be also looked at for additional super-classes of tree-child networks that will be formally introduced in Chapter 6 including reticulation-visible networks and stack-free networks. As will be discussed in Chapter 6 these two super-classes of tree-child networks have interesting properties with regards to specific pairs of embedded base trees and their complement trees which potentially could be useful for this problem.

Chapter 6

Beyond Tree-child Networks

6.1 Introduction

In this chapter important properties found in tree-child networks are isolated and explored in the context of more general network classes. This is done by re-examining difficult network problems that are known to become tractable for the class of tree-child networks and carefully relaxing the corresponding unnecessary structural restraints. Of primary interest are the tree-cover and tree-containment problems. In doing this several network classes, each of which a super-class of tree-child networks, are introduced and explored. For each new network class a formal definition is given in Section 6.2.

Tree-child networks require every non-leaf vertex to have a tree vertex or a leaf as a child. The two restricted relationships between two reticulation vertices in a tree-child network are then being parent and child or siblings. By separating these two restrictions the two new network classes of 'stack-free networks' and 'sibling-free networks' are found. In a stack-free network no reticulation may be the parent of another reticulation and similarly in a sibling-free network no two reticulations may be siblings. Via these two new network classes the two above structural restraints on reticulations in tree-child networks are separated.

Re-examining problems that are known to be tractable for tree-child networks instead for more general network classes isolates the pertinent network constrains. The idea is to tease apart the useful properties of

tree-child networks so that they may be separately applied to more general network classes. The content of this chapter is based on work done jointly with Charles Semple in [SS18].

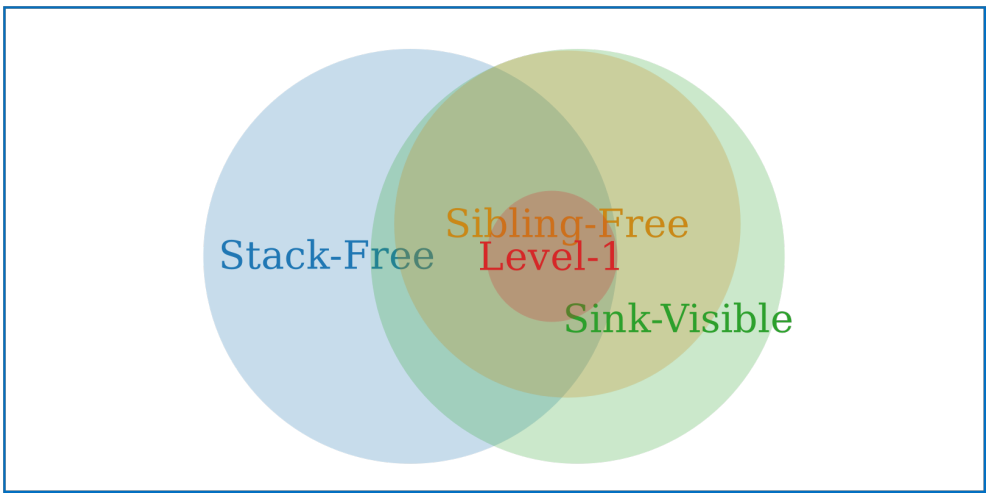

Figure 6.1: Venn Diagram of Phylogenetic Networks.
This Venn diagram shows the interrelationship between sink-visible, sibling-free, stack-free, and level-1 networks. Additionally, the other prominent network classes, reticulation-visible networks and tree-child networks occur as the intersections between pairs of these represented networks. Reticulation-visible networks exist as the intersection between stack-free and sink-visible networks and tree-child networks exist as the intersection between reticulation-visible and sibling-free networks. It may also be noted that level-1, tree-child, and stack-free networks are tree-based. This image was produced using Venn Diagram Maker Online at www.meta-chart.com.

Section 6.3 and Section 6.5 of this chapter primarily address the tree-cover problem, first in the case of stack-free networks and then in the case of other super-classes of tree-child networks. The tree-cover problem asks for a given network $\mathcal{N}$ on X what is the minimum number of embedded phylogenetic trees in $\mathcal{N}$ required to use every vertex and every edge of $\mathcal{N}$.

To directly qualify the question of if evolutionary relationships have a fundamentally tree-like quality, Francis and Steel introduced tree-based networks in [FS15]. One important consequence of a network being tree-based is that there exists an embedded tree in the network, a base tree, that contains or 'covers' every vertex of the network. From [Sem15] and Lemma 5.2.1 it follows that every embedded tree and the

embedded tree produced from the complement switching in a tree-child network together use every vertex and every edge of the network. This means that for any non- tree tree-child network the minimum number of embedded trees in the network required to use every vertex and every edge is at most two.

Since at least [Hei90] networks have often been understood as amalgamations of different gene trees. From this perspective the simplest network that is not a phylogenetic X-tree is one that is, like tree-child networks, an amalgamation of just two phylogenetic X-trees. Extending this property beyond tree-child networks, it is shown that stack-free networks are precisely the class of networks for which there exists two embedded trees that together use every vertex and edge of the network in Proposition 6.3.2. This characterisation leads immediately to a polynomial time algorithm for deciding if an arbitrary network has this property. Particularly surprising, this deceptively simple class is shown to include some universal networks in Proposition 6.5.4. Additionally, the increasingly prominent class of reticulation-visible networks, another super-class of tree-child networks, is re-characterised as a special subclass of stack-free networks in Theorem 6.3.6. Intuitively a network is 'reticulation-visible' if every reticulation in the network is visible. Recent studies of reticulation-visible networks include [BS16] and [GDZ17] and again a formal definition is provided in Section 6.2.

The next main result of this chapter comes from continuing to investigate the tree-cover problem for a new class of networks, "sink-visible networks". Sink-visible networks allow stack-reticulations however the final reticulation of each directed path consisting exclusively of stack-reticulations, the "sink", must be visible. For the class of sink-visible networks the minimum number of phylogenetic networks required to cover every vertex and edge of the network is proven to depend directly on the length of the largest directed path consisting exclusively of stack-reticulations in the network in Theorem 6.5.8.

Finally, in Section 6.6 the last main result of this chapter is given. Like the class of tree-child networks, the tree-containment problem is proven to be solvable in polynomial time for the class of sibling-free networks in Theorem 6.6.5. This is done by adapting a method developed by

Charles Semple to refine the result found in [VISS10] for tree-child networks. As a part of this work a method of simplifying sibling-free networks into networks that are very similar to tree-child networks was used. These simplified sibling-free networks can only be understood as being tree-child networks if the binary requirement of a network is removed. Where this is allowed many of the useful properties of tree-child networks can be directly applied allowing this problem to be solved. Finally in Section 6.7 the results of this chapter are summarised and discussed.

6.2 Preliminaries

In this section the four network classes of stack-free, sibling-free, reticulation-visible and sink-visible networks are formally introduced. The two types of reticulations not allowed in tree-child networks, stack-reticulations and sibling-reticulations, were defined in Chapter 2. Here, these types of reticulations are further explored along with some important network structures that consist of configurations of these reticulations.

Recall from Proposition 2.0.1, that the class of tree-child networks can be characterised as the class of networks, for which there are no stack-reticulations and sibling-reticulations. Furthermore, recall from Proposition 2.0.3, that the class of tree-child networks can also be characterised as the class of networks for which every vertex is visible. In this chapter, these two configurations of reticulations are allowed separately and the requirement of visible vertices is relaxed by varying degrees. From this the above four classes of networks are obtained, each of which a super-class of tree-child networks.

Let $\mathcal{N}$ be a network on X with two reticulations r_1 and r_2. If r_1 is the parent vertex of r_2 then r_1 and r_2 are both stack-reticulations. The edge r_1r_2 joining the two stack-reticulations is a *stack-edge* and the longest directed path containing stack-reticulations and consisting exclusively of stack-edges is a *stack of reticulations*. The size of a stack of reticulations is given by the number of reticulations on that path. The final vertex of a stack of reticulations is a *sink-vertex* and each edge incoming on an element of a stack of reticulations is a *source-edge* of the

stack of reticulations or equivalently of the sink-vertex of the stack of reticulations. An example of a stack of reticulations is given in Fig. 6.2. The directed path consisting of stack-edges is shown in dotted lines and the source-edges are shown in solid lines. Every directed path in the image contains the last reticulation, this reticulation is the sink-vertex.

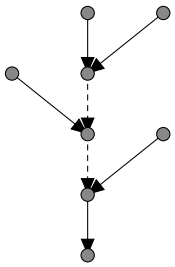

Figure 6.2: Stack of reticulations of size 2.

If r_1 and r_2 are siblings then both r_1 and r_2 are sibling-reticulations. A vertex that is a parent of both r_1 and r_2, such as u or v in Fig. 6.3, is a *sibling-parent*. If for a set of sibling-reticulations there exists a series of up-down paths consisting exclusively of sibling-reticulations and their respective sibling-parents that starts and finishes at the same vertex then the series of up-down paths is collectively a *zigzag cycle*, otherwise known as a *crown*. A simple example of a zigzag cycle is given in Fig. 6.3 where the two up-down paths denoted by solid and dotted lines collectively join sibling-reticulation to sibling-parent to sibling-reticulation to sibling-parent to the first sibling-reticulation again creating a zigzag cycle.

The four network classes of sibling-free, stack-free, reticulation-visible and sink-visible networks are defined as follows. Let $\mathcal{N}$ be a network on X. If there are no stack-reticulations in $\mathcal{N}$ then $\mathcal{N}$ is a *stack-free network*. This class of networks has also been referred to as compressed networks in [HMSW16]. If there are no sibling-reticulations in $\mathcal{N}$ then $\mathcal{N}$ is a *sibling-free network*. If every reticulation in $\mathcal{N}$ is visible then $\mathcal{N}$ is a reticulation-visible network. If every sink-vertex in $\mathcal{N}$ is visible then $\mathcal{N}$ is a *sink-visible network*.

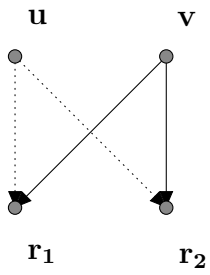

Figure 6.3: Zigzag Cycle.

To conclude this section, the relationships between the above network classes that exist as the intersection of two other above classes are given. First, it will be proven that the class of reticulation-visible networks exists as the intersection between stack-free networks and sink-visible networks. Second, the class of tree-child networks will be proven to exist as the intersection of reticulation-visible networks and sibling-free networks.

Lemma 6.2.1. *The class of reticulation-visible networks exist as the intersection of the classes of stack-free networks and sink-visible networks.*

Proof. Let $\mathcal{N}$ be a reticulation-visible network on X. Every sink in a network is a reticulation. By being a reticulation-visible network it follows immediately that $\mathcal{N}$ is also sink-visible. Furthermore, in a stack of reticulations there exists reticulation vertices that are not visible as each subsequent reticulation allows the previous one to be avoided. Hence $\mathcal{N}$ is also stack-free.

Next, let $\mathcal{N}$ be a stack-free and sink-visible network on X. Because $\mathcal{N}$ is stack-free every reticulation-vertex is a sink. Thus because $\mathcal{N}$ is also sink-visible, $\mathcal{N}$ is a reticulation-visible network. Therefore the class of reticulation-visible networks exists as the intersection between the classes of stack-free networks and sink-visible networks as required. □

Lemma 6.2.2. *The class of tree-child networks exist as the intersection of the classes of reticulation-visible networks and sibling-free networks.*

Proof. Let $\mathcal{N}$ be a tree-child network on X. Because $\mathcal{N}$ is tree-child it immediately follows that $\mathcal{N}$ is sibling-free as the parent vertex of sibling-reticulation vertices has no tree vertex as a child. Furthermore, every vertex in a tree-child network is visible. Hence $\mathcal{N}$ is also a reticulation-visible network.

Next, let $\mathcal{N}$ be a reticulation-visible and sibling free network on X. By being a reticulation-visible network it follows that there are no stack-reticulations in $\mathcal{N}$. It then follows from Proposition 2.0.1 that because $\mathcal{N}$ is a sibling-free and stack-free network, $\mathcal{N}$ is also a tree-child network. Therefore the class of tree-child networks exist as the intersection between the classes of reticulation-visible networks and sibling-free networks as required. □

6.3 Stack-Free Networks

This section begins by identifying that stack-free networks are precisely the class of networks for which the solution to the tree-cover problem is at most two embedded trees. From this result a polynomial time algorithm that decides if an arbitrary network is stack-free will be shown to immediately follow. This allows some networks, typically viewed as complex, to be quickly identified and understood as simply the amalgamation of two underlying phylogenetic X-trees.

Let $\mathcal{N}$ be a network on X that displays a phylogenetic X-tree $\mathcal{T}$. For an embedding $\mathcal{N}_\mathcal{T}$ of $\mathcal{T}$ in $\mathcal{N}$, each vertex and edge in $\mathcal{N}_\mathcal{T}$ is also an element of $\mathcal{N}$. The embedded tree $\mathcal{N}_\mathcal{T}$ is said to *cover* this subset of elements in $\mathcal{N}$. To be more specific, this may also be broken down between vertices and edges and $\mathcal{N}_\mathcal{T}$ be said to *vertex-cover* the vertices of this subset and to *edge-cover* the edges of this subset. Note that if a set T of embedded trees edge-covers a network $\mathcal{N}$ then T also vertex-covers $\mathcal{N}$. For some positive integer k, if the minimum number of embedded trees, and by extension the size of the multi-set of associated displayed phylogenetic X-trees, required to cover $\mathcal{N}$ is k then $\mathcal{N}$ is said to be *k-tree coverable*. The tree-cover problem then asks for a network $\mathcal{N}$ on X that is k-tree coverable what is k.

Lemma 6.3.1. *Let $\mathcal{N}$ be a stack-free network on X. let $E_R(\mathcal{N})$ be the set of reticulation edges in $\mathcal{N}$. There exists a partition $R_1|R_2$ of $E_R(\mathcal{N})$ such that the elements of R_1 and R_2 are both parent vertex and child vertex disjoint.*

Proof. Let $\mathcal{N}$ be a stack-free network on X. let $E_R(\mathcal{N})$ be the set of reticulation edges in $\mathcal{N}$. Suppose that there exists no partition $R_1|R_2$ of $E_R(\mathcal{N})$ such that the elements of R_1 and R_2 are both parent vertex and child vertex disjoint.

Each reticulation-vertex in $\mathcal{N}$ has two parent vertices. By separating each parent vertex pair a partition can be found where each element is child vertex disjoint.

Without loss of generality for each partition where every element is child vertex disjoint, there exists two elements $vr_1, vr_2 \in R_1$ that share a common parent vertex v. As such the alternative incoming edges for the reticulation vertices r_1 and r_2 are elements of R_2. However, this necessarily continues as depicted in Fig. 6.4. In order for no partition, where each element is both parent and child disjoint to exist each additional parent vertex of r_1 and r_2 then also has an additional reticulation child vertex and so on.

This leaves two possibilities. The first, that the zigzag of reticulations continues indefinitely; a contradiction. The second, that the zigzag of reticulations constructs a cycle. In this case an even number of reticulation edges occur with precisely one pair of elements in both R_1 and R_2 having the same parent vertex. By alternating the assignment of each edge in the cycle to R_1 and R_2 so that every second edge is an element of the same set a partition is constructed where each element is parent and child disjoint; a contradiction. Therefore there exists a partition $E_R(\mathcal{N}) = R_1|R_2$ such that the elements of R_1 and R_2 are both parent vertex and child vertex disjoint as required. □

Proposition 6.3.2. *Let $\mathcal{N}$ be a network on X. The network $\mathcal{N}$ is 2-tree coverable if and only if $\mathcal{N}$ contains no stack-reticulations.*

Proof. For the first direction of this proof, let $\mathcal{N}$ be a rooted binary network that contains two embedded trees $\mathcal{N}_{\mathcal{T}_1}$ and $\mathcal{N}_{\mathcal{T}_2}$ that together cover

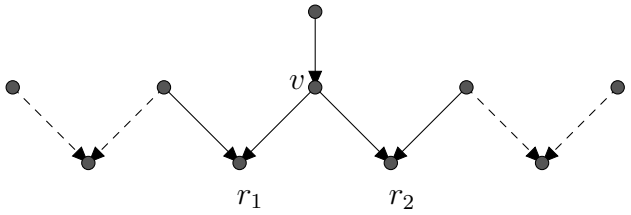

Figure 6.4: A zigzag of reticulations.

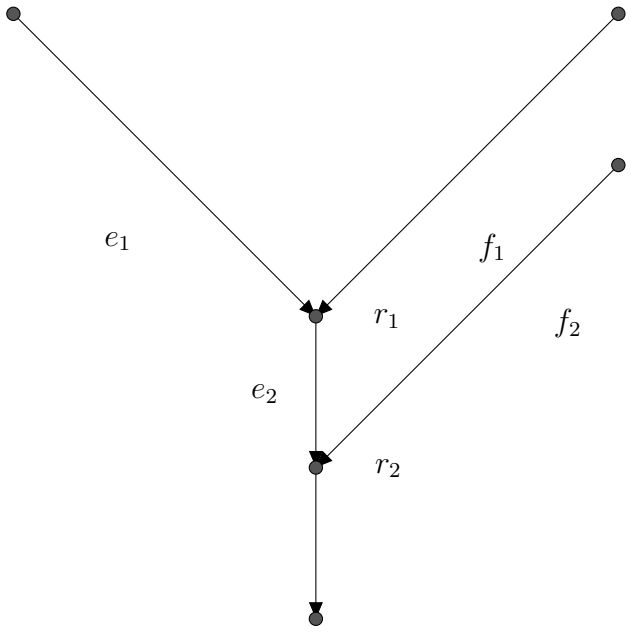

Figure 6.5: Two stack-reticulations r_1 and r_2. The edges e_1 and f_1 and e_2 and f_2 denote the respective incoming edges on r_1 and r_2.

$\mathcal{N}$. Suppose that $\mathcal{N}$ contains at least one pair of stack-reticulations.

Consider a stack of two reticulation vertices r_1 and r_2 in $\mathcal{N}$. As in Fig. 6.5, let the edges incoming on r_1 be e_1 and f_1 and on r_2 are e_2 and f_2 and take e_2 to be the edge joining r_1 and r_2. If the edge e_1 is in $\mathcal{N}_{\mathcal{T}_1}$ then it is clear that f_1 is in $\mathcal{N}_{\mathcal{T}_2}$. However, in order for both $\mathcal{N}_{\mathcal{T}_1}$ and $\mathcal{N}_{\mathcal{T}_2}$ to display phylogenetic X-trees the edge e_2 must then be an element of both embeddings and f_2 then an element of neither. Hence f_2 in $\mathcal{N}$ and $f_2 \notin \mathcal{T}_1, \mathcal{T}_2$; a contradiction.

Thus if there exists two embedded trees $\mathcal{T}_1, \mathcal{T}_2$ that together cover $\mathcal{N}$ then $\mathcal{N}$ contains no stack-reticulation vertices.

For the second direction of this proof, let $\mathcal{N}$ be a stack-free network

on X. Suppose that there exists no two phylogenetic X-trees $\mathcal{T}_1, \mathcal{T}_2$ embedded in $\mathcal{N}$ that together cover $\mathcal{N}$.

It follows from Lemma 6.3.1 that there exists a partition of the reticulation edge set $E_R(\mathcal{N}) = R_1|R_2$ such that the elements of R_1 and R_2 are both parent vertex and child vertex disjoint. Take $E_T(\mathcal{N})$ to be the set of tree edges in $\mathcal{N}$. Now consider the two resulting sets $E_T(\mathcal{N}) \cup R_1$ and $E_T(\mathcal{N}) \cup R_2$.

It is clear that together $E_T(\mathcal{N}) \cup R_1$ and $E_T(\mathcal{N}) \cup R_2$ cover $\mathcal{N}$. Hence without loss of generality, $E_T(\mathcal{N}) \cup R_1$ is not a phylogenetic X-tree and or there exists a vertex v in $\mathcal{N}$ such that $v \notin E_T(\mathcal{N}) \cup R_1$.

By construction for every v in $\mathcal{N}$, the vertex v is an element of $E_T(\mathcal{N}) \cup R_1$. If a path occurred in $E_T(\mathcal{N}) \cup R_1$ that did not terminate at a leaf it would either terminate at a reticulation-vertex or a non-leaf tree vertex. In the first case outgoing from all reticulation vertices in $\mathcal{N}$ is a tree edge $e \in E_T(\mathcal{N}) \cup R_1$; a contradiction. In the second case outgoing from all non-leaf tree vertices in $\mathcal{N}$ are two edges. If one is a tree edge then as above there is a contradiction. If both are reticulation edges then by construction one is an element of $E_T(\mathcal{N}) \cup R_1$; a contradiction.

On the other hand if a path occurred in $E_T(\mathcal{N}) \cup R_1$ that did not begin at the root it would either begin at a non-root non-leaf tree vertex or a reticulation-vertex. Incoming for all non-root tree vertices in $\mathcal{N}$ is a single tree edge. Incoming for all reticulation vertices in $\mathcal{N}$ are two reticulation edges. By construction $E_T(\mathcal{N}) \cup R_1$ contains an incoming edge on the vertex starting the path; a contradiction.

Hence $E_T(\mathcal{N}) \cup R_1$ and $E_T(\mathcal{N}) \cup R_2$ are connected non-cyclic networks that contain each vertex of $\mathcal{N}$ and together cover $\mathcal{N}$. That is there exists two phylogenetic X-trees $\mathcal{T}_1 = E_T(\mathcal{N}) \cup R_1$ and $\mathcal{T}_2 = E_T(\mathcal{N}) \cup R_2$ embedded in $\mathcal{N}$ that together cover $\mathcal{N}$; a contradiction.

Thus if $\mathcal{N}$ contains no stack-reticulation vertices then there exists two phylogenetic X-trees $\mathcal{T}_1, \mathcal{T}_2$ embedded in $\mathcal{N}$ that together cover $\mathcal{N}$. Therefore, the network $\mathcal{N}$ has a covering of two embedded trees if and only if $\mathcal{N}$ contains no stack-reticulation vertices as required. □

An immediate consequence of Proposition 6.3.2 is that there exists a polynomial time algorithm that decides if a network is 2-tree coverable. This is done in the next corollary by systematically checking that no reticulation edge joins two reticulations.

Corollary 6.3.3. *Deciding if an arbitrary phylogenetic network $\mathcal{N}$ is 2-tree coverable can be done in time polynomial on the number of vertices of $\mathcal{N}$*

It follows from Lemma 6.2.1 that the class of reticulation-visible networks is a subclass of stack-free networks. In the next main result, Theorem 6.3.6, it is shown that reticulation-visible networks can moreover be characterised as the special subclass of stack-free networks where any embedded X-tree and the embedded tree containing the complement switching together cover the network. Like in Chapter 5, for an embedded tree in a network, where the complement switching also yields an embedded tree, the two embedded trees are called complement trees.

This means that the important class of reticulation-visible networks then consists exclusively of networks that may, again, be understood as simply the combination of two underlying phylogenetic X-trees. Furthermore, for the class of reticulation-visible networks not only can it be quickly identified that there exists two embedded trees that can together cover the network but on a more practical note, two such embedded trees can be quickly found.

Let $\mathcal{N}$ be a network on X. A *tree-path* in $\mathcal{N}$ is a directed path in $\mathcal{N}$ were every edge in the path is a tree edge. Conversely, a *backwards tree-path* in $\mathcal{N}$ is a tree-path in $\mathcal{N}$ considered running in the reverse direction to that indicated by each edge.

Lemma 6.3.4. *Let $\mathcal{N}$ be a network on X, and let $\mathcal{S}$ be an embedding of a phylogenetic X-tree in $\mathcal{N}$. Let E_1 denote the subset of reticulation edges of $\mathcal{N}$ contained in $\mathcal{S}$. Then $\mathcal{S}$ is the unique embedding in $\mathcal{N}$ of a phylogenetic X-tree containing E_1.*

Proof. Let $\mathcal{S}'$ be an embedding of a phylogenetic X-tree displayed by $\mathcal{N}$ containing each of the edges in E_1. Let F_1 denote the set of tree edges in $\mathcal{N}$ on a backward tree-path starting at either a leaf or a vertex

with an incoming edge in E_1. Observe that if e is a tree edge in F_1, then $\mathcal{S}'$ contains e. But, the edges in $E_1 \cup F_1$ have the property that, for each leaf $x \in X$, there is a directed path from the root of $\mathcal{N}$ to x using only the edges in $E_1 \cup F_1$. Thus $E_1 \cup F_1$ is the edge set of $\mathcal{S}'$, and also the edge set of $\mathcal{S}$. It now follows that $\mathcal{S}' = \mathcal{S}$. □

Lemma 6.3.5. *Let $\mathcal{N}$ be a phylogenetic network on X, and let v be a reticulation in $\mathcal{N}$. If v is not visible, then $\mathcal{N}$ has an embedding of a phylogenetic X-tree avoiding v.*

Proof. Suppose v is not visible. Then, for each $x \in X$, there is a directed path from the root of $\mathcal{N}$ to x that avoids traversing v. Let $\mathcal{S}$ be the embedding of $\mathcal{N}$ induced by the subset of edges of $\mathcal{N}$ in at least one of these paths. Up to degree-two vertices, $\mathcal{S}$ is a phylogenetic network on X. Thus a subset of edges of $\mathcal{S}$ induces an embedding in $\mathcal{N}$ of a phylogenetic X-tree. This embedding avoids v, and so completes the proof of the lemma. □

Theorem 6.3.6. *Let $\mathcal{N}$ be a network on X. The network $\mathcal{N}$ is reticulation-visible if and only if any embedded tree and its complement together cover every element of $\mathcal{N}$.*

Proof. Let $\mathcal{N}$ be a phylogenetic network on X. Suppose that $\mathcal{N}$ is not reticulation-visible. Then, by Lemma 6.3.5, there is an embedding $\mathcal{S}$ in $\mathcal{N}$ of a phylogenetic X-tree that avoids a reticulation, say v. Let E_1 denote the subset of reticulation edges of $\mathcal{N}$ in $\mathcal{S}$. Let $\{E_1', E_2\}$ be a complementary partition of the set of reticulation edges of $\mathcal{N}$, where E_1 is a subset of E_1'. Since $\mathcal{S}$ avoids v, it follows that E_1 is a proper subset of E_1'. It follows from Lemma 6.3.4 that $\mathcal{S}$ is the unique embedding of a phylogenetic X-tree containing E_1, and so there is no embedding in $\mathcal{N}$ of a phylogenetic X-tree containing E_1'. It follows that there exists an embedded tree in $\mathcal{N}$ for which there exists no complement tree that together covers $\mathcal{N}$.

Now suppose that $\mathcal{N}$ is reticulation-visible. Let $\{E_1, E_2\}$ be a complementary partition of the set of reticulation edges in $\mathcal{N}$. First it is shown that $\mathcal{N}$ has an embedding of a phylogenetic X-tree containing the edges in E_1. Let F_1 denote the set of tree edges in $\mathcal{N}$ on a backward

tree-path starting at either a leaf or a vertex that is a tail of an edge in E_1. It is shown that the edges in $E_1 \cup F_1$ induce an embedding $\mathcal{S}_1$ of a phylogenetic X-tree displayed by $\mathcal{N}$.

As E_1 contains exactly one reticulation edge directed into each reticulation, $\mathcal{S}_1$ has no underlying cycles and so, by construction of $\mathcal{S}_1$, there is a (unique) path in $\mathcal{S}_1$ against the direction of the edges from each $x \in X$ to the root ρ of $\mathcal{N}$. Next it is shown that the vertices in $\mathcal{S}_1$ with out-degree zero are vertices in X. If not, then there is a vertex v with out-degree zero in $\mathcal{S}_1$ but with out-degree one or two in $\mathcal{N}$. By construction, v is a reticulation in $\mathcal{N}$. In turn, this implies that v is not visible in $\mathcal{N}$ as there is no $x \in X$ such that every path from ρ to x traverses v; a contradiction. Thus $\mathcal{S}_1$ is an embedding in $\mathcal{N}$ of a phylogenetic X-tree.

Similarly, let F_2 denote the set of tree edges in $\mathcal{N}$ on a backward tree-path starting at either a leaf or a vertex that is the tail of an edge in E_2. Then, as above, the edges in $E_2 \cup F_2$ induces an embedding $\mathcal{S}_2$ of a phylogenetic X-tree displayed by $\mathcal{N}$. Now every reticulation edge of $\mathcal{N}$ is in either $\mathcal{S}_1$ or $\mathcal{S}_2$. To complete the proof, let e be a tree edge in $\mathcal{N}$ that is not an edge in either $\mathcal{S}_1$ or $\mathcal{S}_2$, that is $e \notin F_1 \cup F_2$. Then there is a tree-path consisting of edges not in $F_1 \cup F_2$ from the head of e to a vertex u that is the parent of two reticulations, say v_1 and v_2. But $(u, v_1), (u, v_2) \in E_1 \cup E_2$ and so $e \in F_1 \cup F_2$; a contradiction. Hence the two embedded trees $\mathcal{S}_1$ and $\mathcal{S}_2$ together cover $\mathcal{N}$. It then follows that every embedded tree and its complement together cover $\mathcal{N}$. □

6.4 Questions for Stack-free Networks

In this section several questions that arise for stack-free networks are addressed. To begin with, several smaller results are given for when a switching set yields a base tree in a stack-free network, when a switching and its complement yield two base trees in a reticulation-visible network and when a switching yields a base tree in a tree-based network. These results note the structural requirements of a base tree in a network in terms of switching sets.

Lemma 6.4.1. *Let $\mathcal{N}$ be a stack-free network on X. A switching S on $\mathcal{N}$ yields a base tree if and only if, for all sibling-parents at least one out-going edge is an element of S.*

Proof. Let $\mathcal{N}$ be a stack-free network on X with root ρ. For the first direction of this proof, take S to be a switching on $\mathcal{N}$ such that for all sibling-parents at least one outgoing edge is an element of S. Suppose that S does not yield a base tree of $\mathcal{N}$.

Because S is a switching in $\mathcal{N}$, for each reticulation precisely one of the two incoming edges is an element of S. This implies that there exists no cycles in the network $\mathcal{N}_S$ yielded by S. Additionally, as $\mathcal{N}$ is stack free and each outgoing edge from a reticulation is an element of $\mathcal{N}_S$ for each element ℓ of X there exists at least one path from ρ to ℓ in $\mathcal{N}_S$. Hence $\mathcal{N}_S$ is a phylogenetic X-tree displayed by $\mathcal{N}$.

Consider the reticulation edges not in S. As $\mathcal{N}$ is stack free it is clear that all such edges descend from tree vertices. Moreover, for each such edge there exists a sibling edge. Because S does not yield a base-tree there exists a reticulation edge e_x such that $e_x \notin S$ and its sibling e_y is not an element $\mathcal{N}_S$. However, because S contains at least one outgoing edge from each sibling-parent, the edge e_y is a tree edge. Furthermore, each descendant from e_y must have an outgoing tree edge. Because this tree-path can not be infinitely long it must terminate at a leaf $\ell \in X$. This implies there is no path from ρ to ℓ in $\mathcal{N}_S$; a contradiction.

Hence each edge not in S joins two paths in $\mathcal{N}_S$ a rooted network on X with no cycles. That is S yields a base tree of $\mathcal{N}$; a contradiction. Thus for each switching S on $\mathcal{N}$ such that for all sibling-parents at least one outgoing edge is an element of S, the switching S yields a base-tree of $\mathcal{N}$ as required.

For the second direction of this proof, take S to yield a base tree $\mathcal{T}$ of $\mathcal{N}$. Suppose that there exists a sibling-parent v_{e_i,e_j} with outgoing edges e_i and e_j, such that e_i and $e_j \notin S$. It is clear that the incoming edge on v_{e_i,e_j} is not an element of the embedding $\mathcal{N}_\mathcal{T}$. As such $\mathcal{N}$ can not be constructed from $\mathcal{T}$ by adding only reticulation edges between edges in $\mathcal{T}$. That is $\mathcal{T}$ is not a base tree of $\mathcal{N}$; a contradiction.

Thus if a switching yields a base tree then for each sibling-parent v_{e_i,e_j}

either e_i and/or $e_j \in S$ as required. Therefore a switching S on $\mathcal{N}$ yields a base-tree of $\mathcal{N}$ if and only if, for all sibling-parents at least one outgoing edge is an element of S. □

Corollary 6.4.2. *Let $\mathcal{N}$ be a reticulation-visible network on X. A switching S and complement $\bar{S}$ yield base-trees of $\mathcal{N}$ if and only if, for each sibling-parent v_{e_i,e_j} with outgoing edges e_i and e_j, precisely one of e_i or e_j is an element of S.*

Proof. Let $\mathcal{N}$ be a reticulation-visible network on X with root ρ. Take a switching S and complement $\bar{S}$ in $\mathcal{N}$ such that for each sibling-parent v_{e_i,e_j} with outgoing edges e_i and e_j, precisely one of e_i or e_j is an element of S. Consequently precisely one of e_i or e_j is an element of $\bar{S}$. Hence as $\mathcal{N}$ is also a stack-free network it follows from Lemma 6.4.1 that both S and complement $\bar{S}$ yield base trees of $\mathcal{N}$.

Suppose that both S and complement $\bar{S}$ yield base trees of $\mathcal{N}$ but there exists a sibling-parent v_{e_i,e_j} with outgoing edges e_i and e_j, such that neither e_i or e_j is an element of S. Then again as $\mathcal{N}$ is also a stack-free network it follows from Lemma 6.4.1 that $\bar{S}$ does not yield a base tree of $\mathcal{N}$; a contradiction. Therefore, a switching S and complement $\bar{S}$ yield base-trees of $\mathcal{N}$ if and only if, for each sibling-parent v_{e_i,e_j} with outgoing edges e_i and e_j, precisely one of e_i or e_j is an element of S as required. □

Proposition 6.4.3. *Let $\mathcal{N}$ be a tree-based network on X. A switching S yields a base-tree of $\mathcal{N}$ if and only if S contains at least one outgoing edge from all sibling parents, and all stack-edges.*

Proof. Let $\mathcal{N}$ be a tree-based network on X with root ρ. Take S to be a switching such that S contains at least one outgoing edge from every sibling parent and every edge that joins two reticulation edges. For the first direction of this proof, suppose that S does not yield a base-tree of $\mathcal{N}$.

Label the network yielded by S as $\mathcal{N}_S$. First consider if there exists a leaf $\ell \in X$ such that there is no directed path $P(\rho, \ell)$ in $\mathcal{N}_S$. At least one such path occurs in $\mathcal{N}$. For a path to not be preserved by S that path must contain a reticulation edge $v_i r \notin S$. However, for the

reticulation-vertex r there exists another incoming edge $v_j r \in S$. As $\mathcal{N}$ is tree-based it is clear that a path $P(\rho, v_j)$ also occurs on $\mathcal{N}$. That is for any path from the root to a leaf that is broken in $\mathcal{N}$ there exists an alternative path that is preserved under S. Hence, for all leaves $\ell \in X$ there exists a path from ρ to ℓ in $\mathcal{N}_S$; a contradiction.

It then follows that the cases examined in the proof of Lemma 6.4.1 hold leaving only the consideration that there exists a reticulation edge $r_1 r_2 \notin S$ with a parent vertex that is a reticulation. The edge $r_1 r_2$ should then be an element of S unless both parent vertices of r_2 where reticulations. However this would violate the characterisation of tree-based networks in Corollary 2.11. given by Jetten and Iersal [JvI16]. That is there would exist a zigzag cycle $(o_1, r_1, ..., o_k, r_k, o_{k+1})$, with $k \geq 1$, in which $r_1, ...r_k$ are reticulations, $o_1, ..., o_{k+1}$ are sibling parents and o_1 and o_{k+1} are reticulations as well. Hence a contradiction. Thus S yields a base tree as required.

For the second direction of this proof, suppose there exists a base tree $\mathcal{T}$ such that the switching S' that yields $\mathcal{T}$ does not have the two properties of S. Consider the negation of the first property of S. There exists a sibling parent v such that neither outgoing edges are elements of S'. This implies that the incoming edge on v is not an element of $\mathcal{T}$. As such $\mathcal{N}$ can not be constructed by joining edges in $\mathcal{T}$. That is $\mathcal{T}$ is not a base tree; a contradiction.

Consider the negation of the second property of S. There exists an edge $r_1 r_2$ that joins two reticulation vertices in $\mathcal{N}$ such that $r_1 r_2 \notin S'$. This implies that the vertex $r_1 \notin X$ has in-degree 1 and out-degree 0 in $\mathcal{T}$. As such $\mathcal{T}$ is not a base tree; a contradiction. Thus there exists no such base-tree $\mathcal{T}$.

Therefore, a switching S yields a base-tree of $\mathcal{N}$ if and only if, S contains at least one outgoing edge from all sibling parents and any edge that join two reticulation vertices as required. □

Next, this section addresses two natural questions that arise when considering 2-tree coverable networks. The first, if two embedded trees together cover a network and each displays the same phylogenetic X-tree, what can be known about the network? The second, given two

phylogenetic X-trees and an arbitrary network on X can it be quickly determined if a pair of embeddings of the two trees in the network together cover the network. Because it can be quickly determined if two embedded trees together cover a network, see Corollary 6.3.3, it is equivalent to answer this question for the class of stack-free networks. However, the latter question could only be partially answered for the case of reticulation-visible networks and stack-free networks with the two restrictions of being temporal and containing no zigzag cycles.

Proposition 6.4.4. *Let $\mathcal{N}$ be a network on leaf set X. If $\mathcal{N}$ is covered by two trees and the two trees are the same then the network is either a tree or a level-1 network for which every cycle is a 3-cycle*

Proof. Let $\mathcal{N}$ be a network on leaf set X with a two tree covering consisting of $\mathcal{T}_1$ and $\mathcal{T}_2$ such that $\mathcal{T}_1 = \mathcal{T}_2$. Suppose that $\mathcal{N}$ is not a phylogenetic X-tree or a level-1 network in-which every cycle is a three cycle.

Two embedded trees, $\mathcal{N}_{\mathcal{T}_1}$ and $\mathcal{N}_{\mathcal{T}_2}$ in $\mathcal{N}$, together cover $\mathcal{N}$. It follows from Proposition 6.3.2 that $\mathcal{N}$ is a stack-free network. Consider the following two cases. Case one, there exists two 3-cycles, C_1 and C_2 in $\mathcal{N}$ that share at least one vertex v. Case two, there exists a cycle on least four vertices in $\mathcal{N}$ that is vertex disjoint from any other cycle in $\mathcal{N}$.

In case one, v is either a reticulation or a tree vertex. First, take v to be a reticulation-vertex. Because $\mathcal{N}$ is stack-free neither the parent or child vertices of v are reticulation vertices. Any 3-cycle containing v must consist of both parents of v or one parent and the child of v. However, the child vertex of v is a non-leaf tree vertex or a leaf. This means there can be no three cycle containing both v and its child. Hence v can only be the element of one three cycle which consists of v and its two parent vertices; a contradiction. Second, take v to be a non-leaf tree vertex. Similarly any 3-cycle containing v must consist of both children of v or the parent and one child of v. However, because $\mathcal{N}$ is stack-free if there exists a 3-cycle consisting of the parent of v and one child there can be no 3-cycle consisting of v and its two children. Moreover there can be no two 3-cycles both consisting of the parent

$v'$of v and a child of v. Hence v can only be the element of one three cycle; a contradiction.

In case two, take w to be the sole reticulation-vertex with parent vertices u and v in the cycle C on four or more vertices. Label the not yet labelled children of u, v and w respectively u', v' and w'. Because C is vertex disjoint from all other cycles in $\mathcal{N}$ the child vertices u', v' and w' are each distinct from one and other. In order for the two embedded trees $\mathcal{N}_{\mathcal{T}_1}$ and $\mathcal{N}_{\mathcal{T}_2}$ to cover $\mathcal{N}$ the edge uw must occur in one and vw in the other. However, this creates two different sets of relationships between the child vertices of the elements of C. Note there are at least three child vertices of the elements of C. For $\mathcal{T}_1$ and $\mathcal{T}_2$ to be the same there must be a sink vertex s for which every path from a child vertex of an element of C to a leaf contains s. Because $\mathcal{N}$ is stack-free such a sink can only exists for at most two elements; a contradiction.

Hence if $\mathcal{N}$ has a cycle it is a 3-cycle that is vertex disjoint from all other cycles in $\mathcal{N}$. Therefore, if a network is two tree coverable and the two phylogenetic X-trees are the same then the network is either a phylogenetic X-tree or a level-1 network for which every cycle is a 3-cycle as required. □

Proposition 6.4.5. *Let $\mathcal{N}$ be a normal network on X. If there exists two phylogenetic X-trees, $\mathcal{T}_1$ and $\mathcal{T}_2$ on X that together cover $\mathcal{N}$ then there is only one unique pair of embeddings of $\mathcal{T}_1$ and $\mathcal{T}_2$ that together cover $\mathcal{N}$.*

Proof. Let $\mathcal{N}$ be a normal network on X. Take $\mathcal{T}_1$ and $\mathcal{T}_2$ to be two phylogenetic X-trees that together cover $\mathcal{N}$. It is clear that there exists two embeddings of $\mathcal{T}_1$ and $\mathcal{T}_2$ in $\mathcal{N}$. Moreover, it follows from Lemma 5.2.4 that there can be at most one embedding of each $\mathcal{T}_1$ and $\mathcal{T}_2$ in $\mathcal{N}$. Thus there exists only one unique pair of embeddings of $\mathcal{T}_1$ and $\mathcal{T}_2$ that cover $\mathcal{N}$ as required. □

Lemma 6.4.6. *Let $\mathcal{N}$ be a reticulation-visible network on X with a reticulation edge ur. The network $\mathcal{N} \setminus ur$ is also a reticulation-visible network.*

Proof. Let $\mathcal{N}$ be a reticulation-visible network on X with a reticulation edge uv. Every element of the new network $\mathcal{N} \setminus uv$ is also an element

of $\mathcal{N}$. Hence, similarly every path from the root to a leaf in $\mathcal{N} \setminus uv$ is also a path in $\mathcal{N}$. Additionally, for any reticulation-vertex $r \neq v$ in $\mathcal{N}$ that is visible via a leaf ℓ in X, if uv is an element of a path from r to ℓ then there exists another path from r to ℓ that contains the complement reticulation edge of uv. Hence, the deletion of uv from $\mathcal{N}$ does not result in the bypass or disconnection of any path between a reticulation-vertex and the corresponding leaves that make it visible. As such for every reticulation-vertex r in $\mathcal{N} \setminus uv$ there exists a leaf ℓ in X such that every path from the root of $\mathcal{N} \setminus uv$ to ℓ contains r. Therefore, if $\mathcal{N}$ is a reticulation-visible network then so to is $\mathcal{N} \setminus uv$, as required. □

Proposition 6.4.7. *Let $\mathcal{N}$ be a reticulation-visible network on X with no zigzag cycles on four vertices and $\mathcal{T}_1$ and $\mathcal{T}_2$ two given X-trees. It can be determined in polynomial time if $\mathcal{T}_1$ and $\mathcal{T}_2$ together cover $\mathcal{N}$.*

Proof. Let $\mathcal{N}$ be a reticulation-visible network on X with no zigzag cycles on four vertices and $\mathcal{T}_1$ and $\mathcal{T}_2$ two given phylogenetic X-trees. In [BS16] it was demonstrated that first, it can be determined in polynomial time if a given phylogenetic X-tree is displayed by a reticulation-visible network and second, for $|X| = n$ the network $\mathcal{N}$ has at most $6n - 6$ reticulation edges. Hence it follows from this and Lemma 6.4.6 that it can be checked in polynomial time if $\mathcal{T}_1$ and $\mathcal{T}_2$ are displayed by $\mathcal{N}$ and for each reticulation-vertex r with incoming edges ur and vr in $\mathcal{N}$, if $\mathcal{T}_1$ and $\mathcal{T}_2$ are also displayed by the reticulation-visible networks $\mathcal{N} \setminus ur$ and $\mathcal{N} \setminus vr$. From this it can be identified if each reticulation edge in $\mathcal{N}$ occurs in embeddings of only $\mathcal{T}_1$, only $\mathcal{T}_2$, both or neither.

It is clear that if there exists a reticulation edge that occurs in embeddings of neither $\mathcal{T}_1$ nor $\mathcal{T}_2$ then the two phylogenetic X-trees do not together cover $\mathcal{N}$. Also, if embeddings of either $\mathcal{T}_1$ or $\mathcal{T}_2$ exclusively contain one edge for each reticulation-vertex in $\mathcal{N}$ then it can be checked in polynomial time if this switching and its complement yield embeddings of $\mathcal{T}_1$ and $\mathcal{T}_2$. This leaves only the case where for at least one reticulation-vertex both incoming reticulation edges occur in embeddings of both $\mathcal{T}_1$ and $\mathcal{T}_2$.

Suppose that for a reticulation-vertex r_1 with parent vertices u and v both $\mathcal{N} \setminus ur_1$ and $\mathcal{N} \setminus vr_1$ display both $\mathcal{T}_1$ and $\mathcal{T}_2$. It will be demonstrated

that either r_1 is a part of a 3-cycle or the two phylogenetic X-trees cannot together cover $\mathcal{N}$. By observing the local structure about r_1 it can be immediately identified if r_1 is a part of a 3-cycle. In this case either reticulation edge may be covered by either phylogenetic X-tree. Otherwise it follows from Theorem 6.3.6 that $\mathcal{N}$ is stack free and hence both u and v are tree vertices. Take s and t to then be the sibling vertices of r_1. As r_1 is not an element of a 3-cycle the vertices u and v are distinct from s and t and as r_1 is not an element of a zigzag cycle on four vertices s and t are distinct from each other. Because every reticulation in $\mathcal{N}$ is visible there exists a leaf ℓ_1 in X such that every path from the root to ℓ_1 contains r_1. Finally, label the two subsets of X that descend from s and t as $cl(s)$ and $cl(t)$ respectively.

If neither s, t or one of their descendants is also a reticulation-vertex then it is clear that $\mathcal{N} \setminus ur_1$ and $\mathcal{N} \setminus vr_1$ can not be made to display the same local tree structure. Hence there must exist another reticulation-vertex r_2 and consequently there exists a leaf ℓ_2 in $cl(s) \cup cl(t)$ such that every path from the root to ℓ_2 contains r_2. If $\ell_1 = \ell_2$ then there exists a path from the root to ℓ_1 via the edge us or vt that does not contain r_1. This would mean $\mathcal{N}$ is not reticulation-visible; a contradiction. Hence $\ell_1 \neq \ell_2$.

Now consider $\mathcal{T}_1$. The phylogenetic X-tree $\mathcal{T}_1$ is displayed by both $\mathcal{N} \setminus ur_1$ and $\mathcal{N} \setminus vr_1$. This means that there exists at least two embeddings of $\mathcal{T}_1$ and either there exists a path from u to r_2 in one and v to r_2 in the other or there exists no path from u to r_2 or v to r_2 in any embedding of $\mathcal{T}_1$. In the latter case because $\mathcal{T}_2$ is displayed by both $\mathcal{N} \setminus ur_1$ and $\mathcal{N} \setminus vr_1$ the same situation applies to $\mathcal{T}_2$. However, if there also exists no path from u to r_2 or v to r_2 in any embedding of $\mathcal{T}_2$ then there exists an edge in $\mathcal{N}$ that can not be covered by embeddings of either $\mathcal{T}_1$ or $\mathcal{T}_2$. This leaves the case where there exists at least two embeddings of both $\mathcal{T}_1$ and $\mathcal{T}_2$ where there exists a path from u to r_2 in one and v to r_2 in the other. These paths cannot be direct edges as r_1 is not am element of a zigzag cycle on four vertices. If there is an outgoing edge from the path from u to r_2 in these embeddings then clearly there must be a mirror outgoing edge from the path from v to r_2 in the other that descends to the same leaf set in the embedding. That is there exist two paths now to a third reticulation-vertex r_3. However,

the same argument may then be made about r_2 and r_3 as r_1 and r_2. As X is finite this pattern must stop with a zigzag cycle on four vertices; a contradiction.

Thus either all occurrences where both incoming edges on a reticulation-vertex occur in embeddings of both trees can be attributed to 3-cycles or there the two given phylogenetic X-trees do not together cover the network. Therefore, it can be determined in polynomial time if two given phylogenetic X-trees together cover a reticulation-visible network as required. □

A network $\mathcal{N}$ on X is said to be *temporal* if each vertex u in $\mathcal{N}$ can be assigned a positive real number value $t(u)$ and for each edge vw in $\mathcal{N}$, running from vertex v to w the following two conditions hold. The values $t(v) = t(w)$ if vw is a reticulation edge and $t(v) < t(w)$ if vw is a tree edge.

Proposition 6.4.8. *Let $\mathcal{N}$ be a temporal stack-free network with no zigzag cycles on a leaf set X. It can be determined in polynomial time if two given trees together cover $\mathcal{N}$.*

Proof. Let $\mathcal{N}$ be a temporal stack-free network with no zigzag cycles on a leaf set X with root ρ and r reticulation vertices. Take $\mathcal{T}_1$ and $\mathcal{T}_2$ to be two given phylogenetic trees on leaf sets that are subsets of X. The following is a proof by induction on the sum of the number of leaves $|X|$ and reticulations r in $\mathcal{N}$.

As a base case take $|X| + r = 1$. In this case $\mathcal{N}$, $\mathcal{T}_1$ and $\mathcal{T}_2$ are each a single vertex. As such it is clear that $\mathcal{T}_1$ and $\mathcal{T}_2$ together cover $\mathcal{N}$.

Assume for $|X| + r = k$ that it can be determined in polynomial time if $\mathcal{T}_1$ and $\mathcal{T}_2$ together cover $\mathcal{N}$. Now suppose that when $|X| + r = k + 1$ it can not be determined in polynomial time if $\mathcal{T}_1$ and $\mathcal{T}_2$ together cover $\mathcal{N}$.

Take $P(\rho, \ell)$ to be a path of maximum length from the root to a leaf in $\mathcal{N}$. There then follows two possible cases. First, the parent vertex u of the leaf ℓ is a tree vertex. Because $P(\rho, \ell)$ is of maximum length there can be no path of length grater than one from u to any other leaf. Hence, in this first case ℓ exists as a part of a cherry - specifically

there exists a cherry in $\mathcal{N}$. Second, u is a reticulation-vertex. Because $\mathcal{N}$ is a temporal stack-free network both parent vertices of u are tree vertices and share the same time score. This means that both parent vertices of u either have a leaf as a child or another reticulation-vertex that is itself the parent of a leaf vertex. The latter instance produces a zigzag however as there are no zigzag cycles in $\mathcal{N}$ an end to the zigzag can always be found. Hence in this second case ℓ exists as a part of a reticulated cherry or is descended from a zigzag of reticulations that has an end - specifically there exists a reticulated cherry in $\mathcal{N}$.

If there exists a cherry $\{a, b\}$ in $\mathcal{N}$ then $\mathcal{T}_1$ and $\mathcal{T}_2$ only together cover $\mathcal{N}$ if at least one of them also has the cherry $\{a, b\}$. If this is the case take $\mathcal{N}'$, $\mathcal{T}_1'$ and $\mathcal{T}_2'$ to be the networks obtained by deleting b. By only deleting one leaf from a cherry, $\mathcal{N}'$ is also a temporal stack-free network with no zigzag cycles and $\mathcal{T}_1$ and $\mathcal{T}_2$ together cover $\mathcal{N}$ if and only if $\mathcal{T}_1'$ and $\mathcal{T}_2'$ together cover $\mathcal{N}'$. By the above assumption it can be determined in polynomial time if $\mathcal{T}_1'$ and $\mathcal{T}_2'$ together cover $\mathcal{N}'$. However, it is then clear that by rejoining the leaf b to the phylogenetic X-tree that had it as apart of a cherry and to $\mathcal{N}$ with the leaf a it can also be determined in polynomial time if $\mathcal{T}_1$ and $\mathcal{T}_2$ together cover $\mathcal{N}$; a contradiction.

If there are no cherries then there exists a reticulated cherry $\{a, b\}$ with reticulation c in $\mathcal{N}$. Take b to be the child of c and u to be the parent vertex of a. The phylogenetic trees $\mathcal{T}_1$ and $\mathcal{T}_2$ together cover $\mathcal{N}$ only if at least one of the two has the cherry $\{a, b\}$. If this is the case, without loss of generality take $\mathcal{T}_1$ to have the cherry $\{a, b\}$. Now let $\mathcal{N}'$ be the network obtained by deleting the edge ur and $\mathcal{T}_1'$ be the phylogenetic tree obtained by again deleting b. By only deleting a reticulation edge from a reticulated-cherry, $\mathcal{N}'$ is also a temporal stack-free network with no zigzag cycles and $\mathcal{T}_1$ and $\mathcal{T}_2$ together cover $\mathcal{N}$ if and only if $\mathcal{T}_1'$ and $\mathcal{T}_2$ together cover $\mathcal{N}'$. By the above assumption it can be determined in polynomial time if $\mathcal{T}_1'$ and $\mathcal{T}_2$ together cover $\mathcal{N}'$. However, it is then clear that by rejoining the leaf b the $\mathcal{T}_1$ as apart of a cherry with the leaf a and adding a reticulation edge from the incoming edge on a to the incoming edge on b in $\mathcal{N}$ it can also be determined in polynomial time if $\mathcal{T}_1$ and $\mathcal{T}_2$ together cover $\mathcal{N}$; a contradiction.

Therefore, by induction on the sum of leaves and reticulations in $\mathcal{N}$, it can be determined in polynomial time if two given phylogenetic trees together cover a temporal stack-free network with no zigzag cycles as required. □

It is interesting to note it is because of zigzag cycles that the general problem of, if for two given phylogenetic X-trees and an arbitrary network on X do the two trees together cover the network, is found to be difficult. However, zigzag cycles can occur in reticulation-visible networks. This hints that there should be a way of refining the solution of the general problem further to at least be able to handle specific types of zigzag cycles.

6.5 Counting the Trees that Cover a Network

In this section the tree-cover problem is extended to network classes that are not stack-free. To begin, a minimum number of phylogenetic X-trees to cover a tree-based network is identified.

Lemma 6.5.1. *Let $\mathcal{N}$ be a tree-based network on X with base tree $\mathcal{T}_0$. Take E to be a set of reticulation edges such that for each stack of reticulations in $\mathcal{N}$ precisely one incoming edge on the stack occurs as an element of E. For any such set E there exists an embedded tree $\mathcal{N}_{\mathcal{T}_E}$ where each element of E is an element of $\mathcal{N}_{\mathcal{T}_E}$.*

Proof. Let $\mathcal{N}$ be a tree-based network on X with base tree $\mathcal{T}_0$. Take E to be a set of reticulation edges such that for each stack of reticulations in $\mathcal{N}$ precisely one incoming edge on the stack occurs as an element of E. Suppose that there exists such a set E for which there is no embedded tree $\mathcal{N}_{\mathcal{T}_E}$ where each element of E is an element of $\mathcal{N}_{\mathcal{T}_E}$.

Because $\mathcal{N}$ is tree-based every vertex in $\mathcal{N}$ is also an element of the embedded base tree $\mathcal{N}_{\mathcal{T}_0}$. Additionally, as $\mathcal{N}$ is binary it follows that each element of a stack of reticulations has the same cluster of descendant leaves. There exists a unique directed path from the root to each

leaf in a phylogenetic X-tree hence, the final outgoing edge from each stack is also an element of $\mathcal{N}_{\mathcal{T}_0}$.

Now consider E. For every stack of reticulations one incoming edge is an element of E. All vertices that are joined in $\mathcal{N}$ by an element of E are elements of $\mathcal{N}_{\mathcal{T}_0}$. Thus, for each stack of reticulations $\mathcal{N}_{\mathcal{T}_0}$ may be modified by adding the respective element of E and deleting all bypassed edges into a phylogenetic X-tree embedded in $\mathcal{N}$. That is $\mathcal{N}_{\mathcal{T}_0}$ can be modified into an embedded tree $\mathcal{N}_{\mathcal{T}_E}$ where each element of E is an element of $\mathcal{N}_{\mathcal{T}_E}$; a contradiction.

Therefore, For any such set E there exists an embedded tree $\mathcal{N}_{\mathcal{T}_E}$ where each element of E is an element of $\mathcal{N}_{\mathcal{T}_E}$ as required. □

Proposition 6.5.2. *Let $\mathcal{N}$ be a tree-based network on X with base tree $\mathcal{T}_0$ and largest stack of reticulations S. The minimum number of phylogenetic X-trees required to cover every edge in $\mathcal{N}$ is $|S| + 1$.*

Proof. Let $\mathcal{N}$ be a tree-based network on X with base tree $\mathcal{T}_0$ and largest stack of reticulation vertices S. Take F to be a minimum set of phylogenetic X-trees that covers $\mathcal{N}$. Suppose $|F| \neq |S| + 1$.

For a stack of reticulations no one phylogenetic X-tree can cover more then one incoming edge. As such in order to cover the largest stack of reticulation in $\mathcal{N}$ the relationship $|F| > |S|$ is true.

Consider then the incoming edges on each stack of reticulations in $\mathcal{N}$. Take E to be a set of such edges consisting of precisely one incoming edge for each stack. It follows from Lemma 6.5.1 any set E there exists an embedded tree $\mathcal{N}_{\mathcal{T}_E}$ where each element of E is an element of $\mathcal{N}_{\mathcal{T}_E}$.

As such, a set of phylogenetic X-trees displayed by $\mathcal{N}$ may be chosen so that each phylogenetic X-tree covers one different element from each stack until no more different elements exist. That implies $|F| \leq |S|+1$. Thus $|F| = |S| + 1$; a contradiction. Therefore, The minimum number of phylogenetic X-trees required to cover every edge in $\mathcal{N}$ is $|S| + 1$ as required. □

Next, it is found and demonstrated that there exists a universal temporal tree-based network that can be covered with only two phylogenetic X-trees. This result is a direct consequence of noting that the universal

network found by Zhang in [Zha16] happens to be temporal and stack-free. Because this network happens to be stack-free an unexpected connection can be made between all possible phylogenetic X-trees and just a single pair of phylogenetic X-trees.

Lemma 6.5.3. *There exists a universal temporal network on X.*

Proof. Take a universal tree-based network $\mathcal{N}$ on X of the type constructed by Zhang [Zha16], see Fig. 6.6 and Fig. 6.7. The network $\mathcal{N}$ consists of two parts. Part 1 consists of a 'honeycomb' structure that displays every tree-shape on X, see Fig. 6.6. Part 2 consists of an inter-crossing structure that when joined to the bottom of part 1 allows the display of every possible leaf order of X, see Fig. 6.7.

Suppose that there exists no time function T that can be consistently applied to $\mathcal{N}$. For T to be consistently applied to $\mathcal{N}$ the following two conditions must be met. First, for two vertices u, v in $\mathcal{N}$ joined by a tree edge where u is the parent of v the relationship $T(u) < T(v)$ must hold. Second, for two vertices u, v in $\mathcal{N}$ joined by a reticulation edge the relationship $T(u) = T(v)$ must hold.

Consider first part 1 of Zhang's universal tree-based network. Starting with the root ρ set $T(\rho) = 1$. For each tree-child vertex set its time to its parents plus one. For each reticulation child vertex set its time to its parents. It is clear that a time function can be consistently applied to this part of $\mathcal{N}$.

Consider second the part 2 of Zhang's universal tree-based network. For simplicity set each reticulation-vertex at a different time. Starting from the first reticulation-vertex not in part 1 of Zhang's universal tree-based network set it equal to the final tree vertices of part 1 of Zhang's universal tree-based network. Set each tree vertex at the same time as their descendant reticulation-vertex. It is clear that a time function can be consistently applied to this part of $\mathcal{N}$. Thus, there exists a time function T that can be consistently applied to all of $\mathcal{N}$; a contradiction. Therefore, there exists a universal temporal network on X as required. □

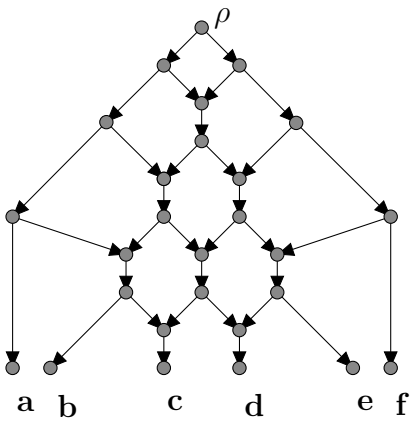

Figure 6.6: Part 1 of Zhang's Universal Tree-Based Networks.

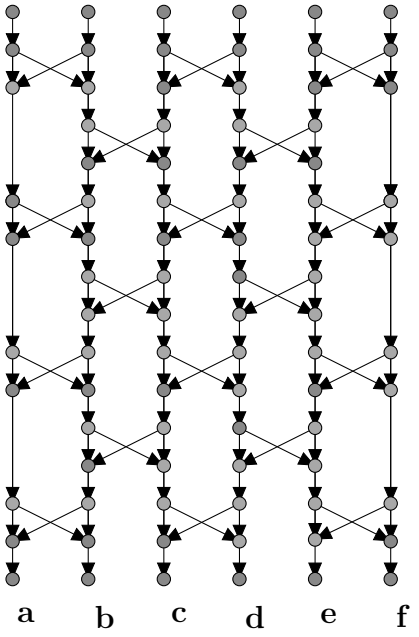

Figure 6.7: Part 2 of Zhang's Universal Tree-Based Networks.

Proposition 6.5.4. *There exists a temporal universal network on X that is 2-tree coverable.*

Proof. Take a universal tree-based network $\mathcal{N}$ on X of the type constructed by Zhang [Zha16]. The network $\mathcal{N}$ consists of two parts. Part 1 consists of a structure that displays every tree shape on X. Part 2 consists of an inter-crossing structure that allows the display of every possible ordering of the leaves in X. It is clear that there exists no stack-reticulations in $\mathcal{N}$. Hence it follows from and Proposition 6.3.2 that $\mathcal{N}$ is 2-tree coverable. Additionally it directly follows from Lemma 6.5.3 that $\mathcal{N}$ is also a temporal universal network on X. Therefore, there exists a temporal universal network on X that is 2-tree coverable as required. □

Finally, this section ends by finding a precise measure for the number of phylogenetic X-trees required to cover a sink-visible network with Theorem 6.5.8. This extends the problem to include networks that are not necessarily tree-based networks whilst maintaining the necessary tractability to produce a definitive outcome.

Lemma 6.5.5. *Let $\mathcal{N}$ be a sink-visible network with root ρ and leaf set X. Each sink vertex v in $\mathcal{N}$ partitions the reticulation edges in $\mathcal{N}$.*

Proof. Let $\mathcal{N}$ be a network with root ρ and leaf set X. Each reticulation-vertex in $\mathcal{N}$ has either a tree vertex or a reticulation-vertex as its only child. That is, every reticulation-vertex is either a sink vertex or a part of a stack of reticulations that terminates at a sink vertex. If a path from ρ to a leaf contains a reticulation edge then it necessarily contains the corresponding reticulation-vertex, its outgoing edge and its child. Hence, if a path from ρ to a leaf contains an element of a stack of reticulations then it also contains the corresponding sink vertex. Thus every reticulation edge is a source edge of a sink vertex.

Suppose that for two sink vertices u and v there exists an edge e such that e is a source edge of both u and v. As such, every path from ρ to a leaf that contains e also contains both u and v. Moreover, there exists two uninterrupted paths of stack-edges $P(e, u)$ and $P(e, v)$. As every edge in $P(e, u)$ and $P(e, v)$ are stack-edges and both paths start

at e, one of $P(e,u)$ and $P(e,v)$ is a sub-path of the other. However both u and v are sink vertices and as such both have an outgoing tree edge. That is one of $P(e,u)$ and $P(e,v)$ is not an uninterrupted path and e then not a source edge of both u and v; a contradiction. Therefore, each sink vertex v in $\mathcal{N}$ partitions the reticulation edges in $\mathcal{N}$ as required. □

Lemma 6.5.6. *Let $\mathcal{N}$ be a sink-visible network with root ρ and leaf set X. Take $S(v)$ to be the set of source edges incoming on a sink vertex v in $\mathcal{N}$. For each phylogenetic X-tree $\mathcal{T}$ embedded in $\mathcal{N}$ there exists precisely one edge $e \in S(v)$ such that e is also an edge in $\mathcal{N}_\mathcal{T}$.*

Proof. Let $\mathcal{N}$ be a sink-visible network with root ρ and leaf set X. It follows from Lemma 6.5.5 that for any sink vertex v in $\mathcal{N}$ there exists a set $S(v)$ of source edges of v. Suppose that there exists a phylogenetic X-tree $\mathcal{T}$ embedded in $\mathcal{N}$ such that $\mathcal{T}$ contains more than one element of $S(v)$ or no elements of $S(v)$.

As $\mathcal{N}$ is sink-visible their exists a leaf ℓ such that every path from ρ to ℓ contains v. As the embedded tree $\mathcal{N}_\mathcal{T}$ consists of a subset of vertices and edges of $\mathcal{N}$, every path in $\mathcal{N}_\mathcal{T}$ is a path in $\mathcal{N}$. Hence the path in $\mathcal{N}_\mathcal{T}$ from ρ ot ℓ must contain v.

First consider the case that $\mathcal{N}_\mathcal{T}$ contains two edges e_1 and $e_2 \in S(v)$. Because $\mathcal{T}$ is an embedded tree there exists a unique path from the root to each of these two edges, $P(\rho, e_1)$ and $P(\rho, e_2)$. As neither e_1 or e_2 are stack-edges neither $P(\rho, e_1)$ or $P(\rho, e_2)$ is a sub-path of the other. Additionally every path from e_1 or e_2 to a leaf in $\mathcal{N}$ also contains v. Hence in $\mathcal{N}_\mathcal{T}$ there exists two paths from the root to v, one containing e_1 and the other containing e_2. However $\mathcal{N}_\mathcal{T}$ is an embedded tree and in a tree there exists only one path from the root to any vertex; a contradiction.

Now consider the case that $\mathcal{N}_\mathcal{T}$ contains no element of $S(v)$. Because v is an element of $\mathcal{N}_\mathcal{T}$ and $v \neq \rho$ there exists an incoming edge e on v in $\mathcal{N}_\mathcal{T}$. The vertex v is a reticulation-vertex in $\mathcal{N}$ and as such e is either a stack-edge or a non-stack-reticulation edge. In the latter case $e \in S(v)$; a contradiction. In the former case the parent vertex of v is a reticulation-vertex in $\mathcal{N}$. The same can be then argued for the incoming

edge on the parent vertex of v. Because ρ is not a reticulation-vertex it is clear that there exists an incoming non-stack-reticulation edge in $\mathcal{N}$ on v or a stack of reticulations that terminates at v that also occur in $\mathcal{N}_\mathcal{T}$. This edge is an element of $S(v)$; a contradiction.

Therefore, for each phylogenetic X-tree $\mathcal{T}$ embedded in $\mathcal{N}$ there exists precisely one edge $e \in S(v)$ such that e is also an edge in $\mathcal{N}_\mathcal{T}$ as required. □

Lemma 6.5.7. *Let $\mathcal{N}$ be a sink-visible network with root ρ and leaf set X. Take $S(v_1)$ and $S(v_2)$ to be the source edge sets of two sink vertices v_1 and $v_2 \in \mathcal{N}$. For any two edges $e_1 \in S(v_1)$ and $e_2 \in S(v_2)$ there exists an embedded tree $\mathcal{N}_\mathcal{T}$ such that e_1 and $e_2 \in \mathcal{N}_\mathcal{T}$.*

Proof. Let $\mathcal{N}$ be a sink-visible network on X with root ρ and at least two sink vertices v_1 and v_2. It follows from Lemma 6.5.5 that there exists two mutually exclusive sets $S(v_1)$ and $S(v_2)$ of source edges incoming on v_1 and v_2. Suppose that for two edges $e_1 \in S(v_1)$ and $e_2 \in S(v_2)$ there exists no embedded tree $\mathcal{N}_\mathcal{T}$ such that e_1 and $e_2 \in \mathcal{N}_\mathcal{T}$.

Take $\mathcal{N}'$ to be an embedded network on X in $\mathcal{N}$ with minimal reticulations such that e_1 and $e_2 \in \mathcal{N}'$. Because $\mathcal{N}'$ contains e_1 and e_2 the network $\mathcal{N}'$ is not a tree. Hence there exists a cycle in $\mathcal{N}'$. Label the two paths that make up the cycle $P(a, b)$ and $P(a, b)'$ where both run from a vertex a to vertex b. In order for both paths to be necessary in $\mathcal{N}'$ each path must contain one of e_1 or e_2. With out loss of generality take $e_1 \in P(a, b)$ and $e_2 \in P(a, b)'$.

As an embedded network $\mathcal{N}'$ consists of the vertex set and a sub-set of the edge set of $\mathcal{N}$. Because $e_1 \in S(v_1)$ and $e_2 \in S(v_2)$ there exists two unique paths $P(e_1, v_1)$ and $P(e_2, v_2) \in \mathcal{N}$ each consisting of stack-edges. Every path in an embedding of $\mathcal{N}$ also occurs in $\mathcal{N}$. Thus as the only paths from e_1 to v_1 and e_2 and v_2 $P(e_1, v_1)$ and $P(e_2, v_2) \in \mathcal{N}'$. The vertex b is then an element of both v_2 $P(e_1, v_1)$ and $P(e_2, v_2)$. However this makes b a stack-reticulation that terminates at both v_1 and v_2. A stack of reticulations can only terminate at one sink vertex and $v_1 \neq v_2$; a contradiction.

Thus there exists no cycle in $\mathcal{N}'$ and furthermore $\mathcal{N}'$ is a tree. Therefore for two edges $e_1 \in S(v_1)$ and $e_2 \in S(v_2)$ there exists an embedded tree

$\mathcal{N}_\mathcal{T}$ in $\mathcal{N}$ such that e_1 and $e_2 \in \mathcal{N}_\mathcal{T}$ as required. □

Theorem 6.5.8. *Let $\mathcal{N}$ be a sink-visible network with root ρ and leaf set X. Take $S(v)$ to be the largest set of non-stack-edges incoming on a sink vertex $v \in \mathcal{N}$. The minimum number of phylogenetic X-trees required to cover $\mathcal{N}$ is equal to $|S(v)|$.*

Proof. Let $\mathcal{N}$ be a sink-visible network with root ρ and leaf set X. It follows from Lemma 6.5.5 that for each sink vertex there exists an exclusive set of source edges. Take $v \in \mathcal{N}$ to be a sink such that $S(v)$ is the set of source edges with the maximum number of elements. It follows from Lemma 6.5.6 that the minimum number of phylogenetic X-trees required to cover every element of $S(v)$ is equal to $|S(v)|$. Additionally from Lemma 6.5.7 it is known that there exists an embedded tree that contains any combination of one element from each set of source edges. Thus the number of phylogenetic X-trees required to cover $\mathcal{N}$ is precisely $|S(v)|$ as required. □

6.6 Sibling-Free Networks

In this section it will be demonstrated that, like tree-child networks, the tree-containment problem can be solved in polynomial time on the size of the leaf set for sibling-free networks. This is done by first identifying that every tree vertex in a sibling-free network is visible. From here the binary requirement on phylogenetic networks is relaxed. Each stack of reticulations will be 'contracted' into a single vertex with an incoming reticulation edge for every source edge. This operation applied to sibling-free networks then produces non-binary tree-child networks. In [VISS10] the tree-containment problem was shown to be solvable in polynomial time for tree-child networks. Subsequently, this result was refined by Charles Semple and shared with me via personal communication. Here the latter method will be mirrored to show that the tree-containment problem can also be solved in polynomial time for sibling-free networks in Theorem 6.6.5.

Lemma 6.6.1. *Let $\mathcal{N}$ be a sibling free network. Every tree vertex in $\mathcal{N}$ has a tree-path to a leaf.*

Proof. Let $\mathcal{N}$ be a sibling free network. Every vertex in $\mathcal{N}$ is a parent to at most one reticulation-vertex. Hence every tree vertex has a tree vertex as a child or is itself a leaf. Thus it can be immediately obtained that for each tree vertex in a sibling-free network there exists a tree-path, (including a trivial tree-path of lenth zero in the case that the vertex is itself a leaf) to a leaf as required. □

Proposition 6.6.2. *A network $\mathcal{N}$ is a sibling-free network if and only if every tree vertex is visible.*

Proof. For the first direction of this proof, let $\mathcal{N}$ be a sibling-free network with root ρ on X. Suppose that there exists a tree vertex u in $\mathcal{N}$ that is not visible. It follows from Lemma 6.6.1 that there exists at least one tree-path $P(u, \ell)$ from u to a leaf ℓ. As $P(u, \ell)$ contains only tree vertices there can be no path in $\mathcal{N}$ from ρ to ℓ that does not contain u. Hence u is visible; a contradiction.

For the second direction of this proof, let $\mathcal{N}$ be a network on X with root ρ such that every tree vertex in $\mathcal{N}$ is visible. Suppose that there exists two reticulation vertices u and v in $\mathcal{N}$ that share a common parent vertex w. Because u and v are reticulation vertices there exists a path from ρ to u and from ρ to v that does not include w. As such there exists no leaf $\ell \in X$ for which every path from ρ to ℓ contains w. Because w has two child vertices w is a tree vertex. Hence there exists a tree vertex in $\mathcal{N}$ that is not visible; a contradiction.

Therefore, a network $\mathcal{N}$ is a sibling-free network if and only if every tree vertex is visible as required. □

For the following few results the requirement that each internal vertex in a network has degree three will be relaxed. This is the requirement that a rooted phylogenetic network be binary and so in general a phylogenetic network without this requirement may be known as non-binary. Specifically in this thesis, a *rooted non-binary phylogenetic network* $\mathcal{N}$ on X is a directed graph with no directed cycles and the following properties.

1. There exists precisely one vertex, the root, with in-degree zero and out-degree either zero or two.

2. The set of vertices with with in-degree one and out-degree zero is X.

3. All other vertices have either in-degree one and out degree two or in-degree equal to or greater than two and out-degree one.

The following operation takes a rooted binary phylogenetic network $\mathcal{N}$ on X and gives a rooted non-binary phylogenetic network. The *stack-contracted* network $sc(\mathcal{N})$ is a rooted non-binary phylogenetic network obtained by contracting every edge uv where u and v are both reticulations in $\mathcal{N}$. It then follows from Proposition 6.6.2 that all stack-contracted sibling-free networks are rooted non-binary tree-child phylogenetic networks. From here on rooted non-binary tree-child phylogenetic networks will be referred to simply as non-binary tree-child or stack-contracted networks and rooted binary networks will continued to be referred to as networks before. Note all phylogenetic X-trees or embedded trees are still considered as binary.

For the next result cherries and reticulated cherries need to be understood for stack-contracted networks. In a stack-contracted network a cherry is, as was defined for rooted binary phylogenetic networks, a pair of sibling leaves. In a stack-contracted network a reticulated cherry is, as was defined for rooted binary phylogenetic networks, a pair of leaves connected by an up-down path of length three that contains precisely one reticulation. It may be noted that in a stack-contracted network several reticulated cherries may share the same reticulation. Additionally, for a network $\mathcal{N}$ on X and $sc(\mathcal{N})$ the stack-contraction of $\mathcal{N}$, if for two leaves a and b in X, the pair $\{a, b\}$ is a cherry in $\mathcal{N}$ then $\{a, b\}$ is a cherry in $sc(\mathcal{N})$. If the pair $\{a, b\}$ is a reticulated cherry in $\mathcal{N}$ then $\{a, b\}$ is a reticulated cherry in $sc(\mathcal{N})$.

Lemma 6.6.3. *Let $sc(\mathcal{N})$ be a stack-contracted sibling-free non-binary network on X where $|X| > 1$. The network $sc(\mathcal{N})$ has either a cherry or a reticulated cherry.*

Proof. Let $sc(\mathcal{N})$ be a stack-contracted sibling-free non-binary network with root ρ on leaf set X. Suppose that $sc(\mathcal{N})$ does not have either a cherry or a reticulated cherry.

Take a path of maximum length from ρ to a leaf $a \in X$ in $sc(\mathcal{N})$. The parent vertex u of a is either a tree vertex or a reticulation-vertex. In the first case u is a tree vertex. As a non-leaf tree vertex u has two child vertices a and another b. Because there exists no cherries in $sc(\mathcal{N})$ b is not a leaf. As such b has a child vertex c and the path from ρ to c is longer then the path from ρ to a; a contradiction.

In the second case u is a reticulation-vertex. As a reticulation-vertex u may have and arbitrary number of parent vertices however one is a tree vertex. Moreover each tree vertex in $sc(\mathcal{N})$ has at least one tree vertex as a child. Because there exists no reticulated cherries in $sc(\mathcal{N})$ every one of these parent vertices of u must have a non-leaf tree vertex child. Take v to be any such vertex. As a non-leaf tree vertex v has two child vertices but because there exists no cherries in $sc(\mathcal{N})$ at least one of which w is not a leaf. If w is not a leaf then w has at least one child vertex x. However the path from ρ to x is then longer than the path form ρ to a; a contradiction. Therefore, the network $sc(\mathcal{N})$ has either a cherry or a reticulated cherry as required. □

Lemma 6.6.4. *Let $\mathcal{N}$ be a sibling-free network on X and $\mathcal{T}$ a phylogenetic X-tree. The network $\mathcal{N}$ displays $\mathcal{T}$ if and only if the stack-contracted network $sc(\mathcal{N})$ also displays $\mathcal{T}$.*

Proof. For the first direction of this proof, let $\mathcal{N}$ be a sibling-free network with root ρ on X. Suppose that $\mathcal{N}$ displays a phylogenetic X-tree $\mathcal{T}$ but the stack contraction $sc(\mathcal{N})$ of $\mathcal{N}$ does not. Because $\mathcal{N}$ displays $\mathcal{T}$ there exists an embedding $\mathcal{N}_\mathcal{T}$ of $\mathcal{T}$ in $\mathcal{N}$. As an embedding every vertex and edge in $\mathcal{N}_\mathcal{T}$ is an element of $\mathcal{N}$. Additionally, every non-stack-reticulation-vertex and edge in $\mathcal{N}$ is an element of the contracted network $sc(\mathcal{N})$. The embedded tree $\mathcal{N}_\mathcal{T}$ either contains a stack of reticulations or it does not. In the former case because there can be no cycles in a tree each stack-reticulation-vertex must be a degree-2 vertex in $\mathcal{N}_\mathcal{T}$. Hence a path between the incoming edge and the outgoing edge of each stack in $\mathcal{N}_\mathcal{T}$ is all that is needed to display $\mathcal{T}$ and such a path exists in $sc(\mathcal{N})$; a contradiction. In the latter case every path in $\mathcal{N}_\mathcal{T}$ is a path in $sc(\mathcal{N})$ thus $sc(\mathcal{N})$ displays $\mathcal{T}$; a contradiction. Therefore if $\mathcal{N}$ displays $\mathcal{T}$ then $sc(\mathcal{N})$ also displays $\mathcal{T}$ as required.

For the second direction of this proof, let $sc(\mathcal{N})$ be a stack-contracted sibling-free network with root ρ on X. Suppose that $sc(\mathcal{N})$ displays a phylogenetic X-tree $\mathcal{T}$ but the non-stack-contracted sibling free network $\mathcal{N}$ does not. Because $sc(\mathcal{N})$ displays $\mathcal{T}$ there exists an embedding $sc(\mathcal{N})_{\mathcal{T}}$ of $\mathcal{T}$ in $sc(\mathcal{N})$. As an embedding every element in $sc(\mathcal{N})_{\mathcal{T}}$ is an element in $sc(\mathcal{N})$. Additionally, every non-stack-reticulation-vertex and edge in $\mathcal{N}$ is an element of the contracted network $sc(\mathcal{N})$. Importantly this includes every edge incoming and the unique edge outgoing of any contracted stack vertices. As such for each incoming edge on a contracted stack vertex there exists a path in $\mathcal{N}_{\mathcal{T}}$ to the unique outgoing edge of that stack. Furthermore, except the first and last vertex each such path consists exclusively of reticulation vertices. Hence for any path in $sc(\mathcal{N})_{\mathcal{T}}$ there exists a path in $\mathcal{N}$ that with the suppression of degree-2 vertices is the same. Thus $\mathcal{N}$ displays $\mathcal{T}$; a contradiction. Therefore if $sc(\mathcal{N})$ displays $\mathcal{T}$ then $\mathcal{N}$ also displays $\mathcal{T}$ as required. □

Theorem 6.6.5. *Let $\mathcal{N}$ be a sibling-free network on X and $\mathcal{T}$ a phylogenetic X-tree. It can be decided whether or not $\mathcal{T}$ is displayed by $\mathcal{N}$ in time polynomial in $|X|$.*

Proof. Let $\mathcal{N}$ be a sibling-free network on X with r reticulation vertices. Let $\mathcal{T}$ be a phylogenetic X-tree. The following is a proof by induction on the sum of the number of leaves $|X|$ and reticulations r in $\mathcal{N}$.

As a base case take $|X| + r = 1$. As such both $\mathcal{N}$ and $\mathcal{T}$ are single vertices. Moreover, $\mathcal{N} = \mathcal{T}$ hence it is clear that it can be decided whether or not $\mathcal{T}$ is displayed by $\mathcal{N}$ in polynomial time as required. Assume that for $|X| + r = k$ it can be decided whether or not $\mathcal{T}$ is displayed by $\mathcal{N}$ in polynomial time. Now suppose that $|X| + r = k + 1$.

It follows from Lemma 6.6.4 that $\mathcal{N}$ displays $\mathcal{T}$ if and only if the stack-contracted network $sc(\mathcal{N})$ also displays $\mathcal{T}$. Because $|X| + r > 1$ it also follows from Lemma 6.6.3 that there exists either a cherry or a reticulated cherry $\{a, b\}$ in $sc(\mathcal{N})$. A cherry or reticulated cherry can be found in polynomial time on $|X|$ by checking for each element of X if its sibling is an element of X or a reticulation-vertex with a child that is an element of X.

First consider the case where $\{a, b\}$ is a cherry in $sc(\mathcal{N})$. Then $\{a, b\}$

is a cherry in each phylogenetic X-tree displayed by $sc(\mathcal{N})$. Thus, if a, b is not a cherry in $\mathcal{T}$, then $sc(\mathcal{N})$ does not display $\mathcal{T}$. If a, b is a cherry in $\mathcal{T}$, then reduce $\{a, b\}$ to a single leaf ℓ in $sc(\mathcal{N})$ and $\mathcal{T}$ to obtain a new network $sc(\mathcal{N}')$ and a new phylogenetic X-tree $\mathcal{T}'$. Note that $sc(\mathcal{N}')$ is a sibling-free network on a $|X| - 1$ leaves and r reticulations. Furthermore, it is easily checked that $sc(\mathcal{N})$ displays $\mathcal{T}$ if $sc(\mathcal{N}')$ displays $\mathcal{T}'$.

Next consider the case where $\{a, b\}$ is a reticulated cherry in $sc(\mathcal{N})$. Without loss of generality take the parent vertex u of a to be the reticulation-vertex. Let v be the parent vertex of b. Note that v is also a parent vertex of u. As u is a reticulation-vertex in a stack-contracted network u may have an arbitrary number of parent vertices. Let W be the set of parent vertices of u excluding v. Now further break the problem into two more sub-cases where either $\{a, b\}$ is a cherry in $\mathcal{T}$ or not.

If $\{a, b\}$ is a cherry in $\mathcal{T}$, then delete each edge $w_i u$ for all $w_i \in W$ and suppress all resulting degree-2 vertices in $sc(\mathcal{N})$ to obtain the new network $sc(\mathcal{N}')$. Because $sc(\mathcal{N})$ is a sibling-free network, each $w_i \in W$ is a tree vertex. As such, it follows from Lemma 6.6.1 that there exists a tree-path to a leaf $\ell_i \neq a$ for each w_i. Hence $sc(\mathcal{N}')$ is a stack compacted sibling-free network on X. Here every phylogenetic X-tree displayed by $sc(\mathcal{N}')$ has the cherry $\{a, b\}$ and the sum of leaves and reticulations in $sc(\mathcal{N}')$ is less than $k + 1$.

It is easily observed that if $sc(\mathcal{N}')$ displays $\mathcal{T}$, then so to does $sc(\mathcal{N})$. In the other direction, if $sc(\mathcal{N})$ displays $\mathcal{T}$ take the up-down path $P(a : b)$ in an embedding $sc(\mathcal{N})_{\mathcal{T}}$ of $\mathcal{T}$ in $sc(\mathcal{N})$. If $P(a : b)$ contains vu then $sc(\mathcal{N}')$ also displays $\mathcal{T}$. If $P(a : b)$ does not contain vu then as the parent vertex of b the vertex v is an element of $P(a : b)$. By keeping the directed path from the peak vertex of P to b via v and replacing all other edges in P with vu in $sc(\mathcal{N})_{\mathcal{T}}$ a new embedding $sc(\mathcal{N})_{\mathcal{T}}$ of $\mathcal{T}$ in $sc(\mathcal{N})$ is obtained. Every element of $sc(\mathcal{N})_{\mathcal{T}}$ is also an element of $sc(\mathcal{N}')$. Thus if $sc(\mathcal{N})$ displays $\mathcal{T}$, then so to does $sc(\mathcal{N}')$.

If $\{a, b\}$ is not a cherry in $\mathcal{T}$, then delete the edge vu in $sc(\mathcal{N})$ to obtain the new network $sc(\mathcal{N}')$. Again it is clear that $sc(\mathcal{N}')$ is a stack compacted sibling-free network on X and the sum of leaves and

reticulations in $sc(\mathcal{N}')$ is less than $k+1$. Because no embedding of $\mathcal{T}$ in $sc(\mathcal{N})$ contains the edge vu it is easily observed that $sc(\mathcal{N})$ displays $\mathcal{T}$ if and only if $sc(\mathcal{N}')$ displays $\mathcal{T}$.

Thus by induction on the sum of leaves and reticulations in $\mathcal{N}$, if $\{a, b\}$ is a cherry or a reticulated cherry in $\mathcal{N}$ then it can be decided whether or not $\mathcal{T}$ is displayed by $sc(\mathcal{N}')$ in polynomial time. Because contracting a cherry or deleting an edge can be done in constant time and $\mathcal{N}$ displays $\mathcal{T}$ if and only if $sc(\mathcal{N}')$ displays $\mathcal{T}$ it can be decided whether or not $\mathcal{T}$ is displayed by $\mathcal{N}$ in polynomial time as required. □

Spanning tree. Let $\mathcal{N}$ be a network on X. A phylogenetic tree $\mathcal{T}$ is a *spanning tree* in $\mathcal{N}$ if $\mathcal{T}$ is a connected network with no cycles and every vertex in $\mathcal{N}$ is a vertex in $\mathcal{T}$.

Finally, this section concludes by considering spanning trees on stack-contracted sibling-free networks. Again similar to tree-child networks a phylogenetic X-tree is found to be a spanning tree of a stack-contracted sibling-free network if and only if the phylogenetic X-tree is an embedded tree of the network.

Proposition 6.6.6. *Let $\mathcal{N}$ be a stack-contracted sibling-free network on X. A phylogenetic tree $\mathcal{T}$ is a spanning tree in $\mathcal{N}$ if and only if $\mathcal{T}$ is an embedded tree on X in $\mathcal{N}$.*

Proof. For the first direction of this proof, let $\mathcal{N}$ be a stack-contracted sibling-free network on a leaf set X. Suppose that there exists a spanning tree $\mathcal{T}$ in $\mathcal{N}$ but $\mathcal{T}$ is not a an embedded tree on X in $\mathcal{N}$.

Because $\mathcal{T}$ is a spanning tree every vertex in $\mathcal{N}$ is an element of $\mathcal{T}$. As such every leaf in $\mathcal{N}$ is a leaf in $\mathcal{T}$. However $\mathcal{T}$ does not occur on the leaf set X. This means that there exists a non-leaf vertex v in $\mathcal{N}$ that is a leaf in $\mathcal{T}$. It follows from Proposition 6.6.2 that the vertex v is visible in $\mathcal{N}$. Hence there exists a leaf ℓ in X such that every path from the root of $\mathcal{N}$ to ℓ passes through v. Because ℓ is an element of $\mathcal{T}$ the path from v to ℓ must also be an element of $\mathcal{T}$. There then exists no such vertex v that is a non-leaf vertex in $\mathcal{N}$ and a leaf vertex in $\mathcal{T}$. Thus $\mathcal{T}$ is an embedded phylogenetic X-tree in $\mathcal{N}$; a contradiction.

For the second direction of this proof, let $\mathcal{N}$ be a stack-contracted sibling-free network on a leaf set X. Suppose that there exists an embedded phylogenetic X-tree $\mathcal{T}$ in $\mathcal{N}$ but $\mathcal{T}$ is not a spanning tree in $\mathcal{N}$.

Because $\mathcal{T}$ is not a spanning tree there exists a vertex v in $\mathcal{N}$ that is not an element of $\mathcal{T}$. However it again follows from Proposition 6.6.2 that the vertex v is visible in $\mathcal{N}$. Hence there exists a leaf ℓ in X such that every path from the root of $\mathcal{N}$ to ℓ passes through v. Because $\mathcal{T}$ is a phylogenetic X-tree the leaf ℓ is an element of $\mathcal{T}$. The path from v to ℓ then must also be an element of $\mathcal{T}$. There then exists no such vertex v that is an element of $\mathcal{N}$ but not $\mathcal{T}$. Thus $\mathcal{T}$ is a spanning tree of $\mathcal{N}$; a contradiction. Therefore, a phylogenetic X-tree $\mathcal{T}$ is a spanning tree in a stack-contracted sibling-free network $\mathcal{N}$ if and only if $\mathcal{T}$ is an embedded phylogenetic X-tree in $\mathcal{N}$ as required. □

Proposition 6.6.7. *Let $\mathcal{N}$ be a stack-contracted sibling-free network on a leaf set X. Every embedded tree in $\mathcal{N}$ is a base-tree of $\mathcal{N}$.*

Proof. Let $\mathcal{N}$ be a stack-contracted sibling-free network on a leaf set X. Suppose that there exists an embedded tree $\mathcal{T}$ in $\mathcal{N}$ that is not a base-tree of $\mathcal{N}$.

It follows from Proposition 6.6.6 that $\mathcal{T}$ is a spanning tree and as such every vertex in $\mathcal{N}$ is a vertex in $\mathcal{T}$. Every edge in $\mathcal{N}$ that is not in $\mathcal{T}$ must join two vertices that are in $\mathcal{T}$. Hence by adding only edges that join two paths in $\mathcal{T}$ the network $\mathcal{N}$ can be obtained. That is $\mathcal{T}$ is a base-tree of $\mathcal{N}$; a contradiction. Therefore, if $\mathcal{N}$ is a stack-contracted sibling-free network then every embedded tree in $\mathcal{N}$ is a base-tree of $\mathcal{N}$ as required. □

6.7 Summary

In this chapter two specific results known for tree-child networks were explored for more general classes of networks. Of interest, was the question of how many embedded trees are necessary to cover every edge of a network and the tree-containment problem. By relaxing the restrictions on tree-child networks, more general classes of networks

where found for which there remained sufficient tractability for these questions to also be answered.

The class of stack-free networks was characterised as the class of networks for which every network has two embedded trees that together cover the network. Here a polynomial time algorithm was found that could identify if an arbitrary network could be considered as simply a combination of two phylogenetic X-trees. Additionally, a new characterisation of reticulation-visible networks was identified. That is reticulation-visible networks where shown to be precisely the class of networks for which any embedded tree and its complement may be chosen to cover the network.

The class of sibling-free networks was demonstrated along the same lines as tree-child networks as a class for which the tree-containment problem is solvable in polynomial time. Of particular interest for future work, the method of contracting stacks of reticulations used to obtain this proof appears to give a natural stepping stone for transitioning from binary networks to non-binary networks.

Chapter 7

Conclusion

To conclude, the purpose of this work was to explore the important, but poorly understood, relationship between classes of phylogenetic networks and the underlying sets of trees that they display. If a network on X is to be viewed as an amalgamation of individual phylogenetic X-trees, then the interplay between the structural properties of those trees and how they are allowed to be combined in to a network, consequently impacts the emergent properties of the network. As gene trees are generally viewed as an attainable and reasonable representation of genetic history, a detailed understanding of the different structural consequences that result from the requirements of different classes of networks is essential.

Chapters 3, 4, and 5 all centre on the network-capture problem. The ambition was to find necessary and sufficient conditions for there to exist a tree-child network that displays a given set of phylogenetic X-trees. In addition in the positive case it was hoped that a polynomial time algorithm would be found that can construct such a network.

Chapter 3 addressed the network-capture problem with the highly structured network class of level-1 networks. Here a polynomial time algorithm was found that, when possible, constructed the unique, up to a certain structural property, level-1 network that displays a given set of phylogenetic X-trees with the minimum number of reticulation vertices. This showed that for certain tree-child networks the network capture problem is solvable in polynomial time. Additionally, this gave

an outline for how the problem might be addressed for more complicated classes of networks.

Chapter 4 compared the properties of level-1 networks and tree-child networks. In the first part of this chapter, where $\mathcal{T}_1$ and $\mathcal{T}_2$ are a pair of phylogenetic X-trees and $S_{\mathcal{T}_1,\mathcal{T}_2}$ is the set of tree-child networks that displays the pair with minimal reticulations, it was demonstrated that though the additional versatility of tree-child networks means $S_{\mathcal{T}_1,\mathcal{T}_2} \neq \emptyset$, it also means $|S_{\mathcal{T}_1,\mathcal{T}_2}| \neq 1$ for all pairs of trees. Hence, the network capture problem of a given set T, where $|T| < 2$, cannot be solved in the case of tree-child networks by simply considering pairs of phylogenetic X-trees in T in a bootstrapping process. The second part of the chapter continued looking at other limits to what phylogenetic X-trees may be displayed by a tree-child network. This led to the surprising result that there exists a set of three phylogenetic X-trees that cannot be displayed by any tree-child network. The particular properties responsible for this where generalised and made quickly identifiable for any given set of phylogenetic X-trees.

Chapter 5 returned to the network-capture problem for tree-child networks with the benefit of two assumptions to provide necessary tractability. The first, that for the given set of phylogenetic X-trees there exists a tree-child network that displays every element of the set. The second, that this tree-child network contains no short-cut edges. In other words, the network is a normal network on X. Together these two assumptions reduce the problem to faithfully reconstructing a known normal network from the set of phylogenetic X-trees it displays, or a subset thereof. For this problem the properties found in Chapter 3 for level-1 networks where exploited and a fast method for reconstructing a normal network from a linear subset of the phylogenetic X-trees it displays was found.

Chapter 6 looked to extend useful results and properties of tree-child networks to more generalised classes of networks. This began with the characterisation of the class of stack-free networks as the class of networks for which there exists two embedded trees that together cover the network. Next a new characterisation of the class of reticulation-visible networks as the class of networks for which any embedded tree

and its complement together cover the network. Despite the obvious complexity, it was also demonstrated that the underlying tree structure of even some universal networks are comprised of simply two phylogenetic X-trees combined together. Additionally, with some restrictions on the network it was found that given two phylogenetic X-trees and a network it could be quickly determined if embeddings of the two trees could cover the network. These restrictions where that the network be reticulation-visible or be temporal and have no zigzag cycles. Interestingly it might be noted that zigzag cycles can occur in reticulation-visible networks. This hints that a more general solution might be possible. Finally, looking at the other restricted reticulation-vertex relationship in tree-child networks, the tree-containment problem was found, in a similar manner as with tree-child networks, to be solvable in polynomial time for the class of sibling-free networks.

To expand on this future work might look to answer the following questions. How many ways can two phylogenetic X-trees be combined into a tree-child network? It is well-known that for any two phylogenetic X-trees there exists a tree-child network that displays both trees. If the number of such networks could be counted and found to grow slowly with the size of the leaf set then a solution to the network-capture problem, similar to that for level-1 networks, might be applicable to tree-child networks as well. Can any insight into what tree-child networks can and cannot display be found from looking at a distance between embedded trees in terms of uni-cyclic networks and base-trees? It is known that every embedded tree in a tree-child network is a base-tree of the network. However additionally, for each base-tree in a tree-child network there is a set of other embedded trees that differ from the base-tree by the use of precisely one reticulation edge, the cover set. Any two trees in a tree-child network might be compared in terms of steps from the first phylogenetic X-tree to its cover-tree to the cover-tree's cover-tree and onwards to the second phylogenetic X-tree. This may find that some embedded trees are more central than others or give parameters that identify sets of phylogenetic X-trees that are more readily captured by a network than others. Finally, by contracting stacks of reticulations in sibling-free networks can other useful results for tree-child networks be replicated for non-binary networks?

Bibliography

[AG05] Quentin D Atkinson and Russell D Gray. Curious parallels and curious connections—phylogenetic thinking in biology and historical linguistics. *Systematic Biology*, 54(4):513–526, 2005.

[All70] Frances E Allen. Control flow analysis. In *ACM Sigplan Notices*, volume 5, pages 1–19. ACM, 1970.

[ASSU81] Alfred V. Aho, Yehoshua Sagiv, Thomas G. Szymanski, and Jeffrey D. Ullman. Inferring a tree from lowest common ancestors with an application to the optimization of relational expressions. *SIAM Journal on Computing*, 10(3):405–421, 1981.

[BGMS05] Mihaela Baroni, Stefan Grünewald, Vincent Moulton, and Charles Semple. Bounding the number of hybridisation events for a consistent evolutionary history. *Journal of Mathematical Biology*, 51(2):171–182, 2005.

[BS15] Magnus Bordewich and Charles Semple. Determining phylogenetic networks from inter-taxa distances. *Journal of Mathematical Biology*, pages 1–21, 2015.

[BS16] Magnus Bordewich and Charles Semple. Reticulation-visible networks. *Advances in Applied Mathematics*, 78:114–141, 2016.

[BSS05] Mihaela Baroni, Charles Semple, and Mike Steel. A framework for representing reticulate evolution. *Annals of Combinatorics*, 8(4):391–408, 2005.

[BvIJ+13] Eric Bapteste, Leo van Iersel, Axel Janke, Scot Kelchner, Steven Kelk, James O McInerney, David A Morrison, Luay Nakhleh, Mike Steel, Leen Stougie, et al. Networks: expanding evolutionary thinking. *Trends in Genetics*, 29(8):439–441, 2013.

[CLS14] Paul Cordue, Simone Linz, and Charles Semple. Phylogenetic networks that display a tree twice. *Bulletin of Mathematical Biology*, 76(10):2664–2679, 2014.

[CRV09] Gabriel Cardona, Francesc Rossello, and Gabriel Valiente. Comparison of tree-child phylogenetic networks. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 6(4):552–569, 2009.

[Dar37] Charles Darwin. Charles darwin notebook b. https://en.wikipedia.org, 1837. Accessed: 24/10/2018.

[DDBR09] James H Degnan, Michael DeGiorgio, David Bryant, and Noah A Rosenberg. Properties of consensus methods for inferring species trees from gene trees. *Systematic Biology*, 58(1):35–54, 2009.

[DM06] Tal Dagan and William Martin. The tree of one percent. *Genome Biology*, 7(10):118, 2006.

[Doo99] W Ford Doolittle. Phylogenetic classification and the universal tree. *Science*, 284(5423):2124–2128, 1999.

[Eis08] Leonard Eisenberg. The tree of life. https://www.evogeneao.com/learn/tree-of-life, 2008. Accessed: 24/10/2018.

[FS15] Andrew R Francis and Mike Steel. Which phylogenetic networks are merely trees with additional arcs? *Systematic Biology*, 64(5):768–777, 2015.

[GDZ17] Andreas DM Gunawan, Bhaskar DasGupta, and Louxin Zhang. A decomposition theorem and two algorithms for reticulation-visible networks. *Information and Computation*, 252:161–175, 2017.

[GVIK+16] Philippe Gambette, Leo Van Iersel, Steven Kelk, Fabio Pardi, and Celine Scornavacca. Do branch lengths help to locate a tree in a phylogenetic network? *Bulletin of Mathematical Biology*, 78(9):1773–1795, 2016.

[HBMR04] Christopher Howe, Adrian Barbrook, Linne Mooney, and Peter Robinson. Parallels between stemmatology and phylogenetics. *Studies in Stemmatology II*, 2:3, 2004.

[Hei90] Jotun Hein. Reconstructing evolution of sequences subject to recombination using parsimony. *Mathematical Biosciences*, 98(2):185–200, 1990.

[HMSW16] Katharina T Huber, Vincent Moulton, Mike Steel, and Taoyang Wu. Folding and unfolding phylogenetic trees and networks. *Journal of Mathematical Biology*, 73(6-7):1761–1780, 2016.

[HRB+09] Daniel H Huson, Regula Rupp, Vincent Berry, Philippe Gambette, and Christophe Paul. Computing galled networks from real data. *Bioinformatics*, 25(12):i85–i93, 2009.

[HvIKS11] Katharina T Huber, Leo van Iersel, Steven Kelk, and Radoslaw Suchecki. A practical algorithm for reconstructing level-1 phylogenetic networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 8(3):635–649, 2011.

[JS06] Jesper Jansson and Wing-Kin Sung. Inferring a level-1 phylogenetic network from a dense set of rooted triplets. *Theoretical Computer Science*, 363(1):60–68, 2006.

[JvI16] Laura Jetten and Leo van Iersel. Nonbinary tree-based phylogenetic networks. *IEEE/ACM transactions on computational biology and bioinformatics*, 2016.

[KNTX08] Iyad A Kanj, Luay Nakhleh, Cuong Than, and Ge Xia. Seeing the trees and their branches in the network is hard. *Theoretical Computer Science*, 401(1-3):153–164, 2008.

[Koo15] Eugene V Koonin. The turbulent network dynamics of microbial evolution and the statistical tree of life. *Journal of molecular evolution*, 80(5-6):244–250, 2015.

[LS19] Simone Linz and Charles Semple. Attaching leaves and picking cherries to characterise the hybridisation number for a set of phylogenies. *Advances in Applied Mathematics*, 105:102–129, 2019.

[MSW15] Colin McDiarmid, Charles Semple, and Dominic Welsh. Counting phylogenetic networks. *Annals of Combinatorics*, 19(1):205–224, 2015.

[OWVIM16] James Oldman, Taoyang Wu, Leo Van Iersel, and Vincent Moulton. Trilonet: piecing together small networks to reconstruct reticulate evolutionary histories. *Molecular Biology and Evolution*, 33(8):2151–2162, 2016.

[Sch70] E Schröder. Vier combinatorische probleme. *Zeitschrift für Mathematik und Physik 15*, pages 361–376, 1870.

[Sem15] Charles Semple. Phylogenetic networks with every embedded phylogenetic tree a base tree. *Bulletin of Mathematical Biology*, 2015.

[SS18] Charles Semple and Jack Simpson. When is a phylogenetic network simply an amalgamation of two trees? *Bulletin of Mathematical Biology*, pages 1–11, 2018.

[Ste16] Mike Steel. *Phylogeny: discrete and random processes in evolution*. SIAM, 2016.

[SWG05] Yun S Song, Yufeng Wu, and Dan Gusfield. Efficient computation of close lower and upper bounds on the minimum number of recombinations in biological sequence evolution. *Bioinformatics*, 21(suppl_1):i413–i422, 2005.

[VIK11] Leo Van Iersel and Steven Kelk. When two trees go to war. *Journal of Theoretical Biology*, 269(1):245–255, 2011.

[VIKK+08] Leo Van Iersel, Judith Keijsper, Steven Kelk, Leen Stougie, Ferry Hagen, and Teun Boekhout. Constructing level-2 phylogenetic networks from triplets. In *Annual International Conference on Research in Computational Molecular Biology*, pages 450–462. Springer, 2008.

[VISS10] Leo Van Iersel, Charles Semple, and Mike Steel. Locating a tree in a phylogenetic network. *Information Processing Letters*, 110(23):1037–1043, 2010.

[Wil07] Stephen J Willson. Unique determination of some homoplasies at hybridization events. *Bulletin of Mathematical Biology*, 69(5):1709–1725, 2007.

[Wil11] Stephen Willson. Regular networks can be uniquely constructed from their trees. *IEEE/ACM Transactions on Com-*

putational Biology and Bioinformatics (TCBB), 8(3):785–796, 2011.

[Wit13] Ludwig Wittgenstein. *Tractatus logico-philosophicus*. Routledge, 2013.

[WZZ01] Lusheng Wang, Kaizhong Zhang, and Louxin Zhang. Perfect phylogenetic networks with recombination. *Journal of Computational Biology*, 8(1):69–78, 2001.

[Zha16] Louxin Zhang. On tree-based phylogenetic networks. *Journal of Computational Biology*, 23(7):553–565, 2016.